# COMPUTER SIMULATION OF THE UNDERWATER DETECTION OF SPERM WHALES USING MODIFICATIONS OF LINE TRANSECT THEORY

A MAJOR REPORT
BY
ANNE M. WERNER

Submitted to Dr. Robert E. Randall and Dr. Robert H. Benson of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING

August 1993

Major Subject: Ocean Engineering

| AD NUMBER | DATE | DTIC ACCESSION NOTICE |
|---|---|---|
| 1. REPORT IDENTIFYING INFORMATION<br><br>A. ORIGINATING AGENCY<br>NAVAL POSTGRADUATE SCHOOL<br>MONTEREY, CA 93943-5100<br><br>B. REPORT TITLE AND/OR NUMBER<br>COMPUTER SIMULATION OF THE UNDERWATER DETECTION OF SPERM WHALES USING MODIFICATIONS OF LINE TRANSECT THEORY<br><br>C. MONITOR REPORT NUMBER<br>BY: ANNE M. WERNER THESIS, TEXAS A&M UNIVERSITY AUG.93<br><br>D. PREPARED UNDER CONTRACT NUMBER<br>N00123-89-G-0598<br><br>2. DISTRIBUTION STATEMENT<br><br>A | | **REQUESTER:**<br>1. *Put your mailing address on reverse of form.*<br>2. *Complete items 1 and 2.*<br>3. *Attach form to reports mailed to DTIC.*<br>4. *Use unclassified information only.*<br>5. *Do not order document for 6 to 8 weeks.*<br><br>**DTIC:**<br>1. *Assign AD Number.*<br>2. *Return to requester.* |

DTIC Form 50 DEC 91 PREVIOUS EDITIONS ARE OBSOLETE

## ABSTRACT

The underwater acoustical detection of sperm whales and the estimation of their population density in the ocean environment is simulated using QBASIC program code. The program is designed to imitate the known behavior of sperm whales and the randomness of the natural environment encountered in the field. A ship-towed linear hydrophone array is modeled for detection of the whales. The program uses the basic theory of line transect sampling. The program can be run in either two-dimensions or three dimensions and is intended to be used in comparison and testing of two- and three-dimensional line transect theory.

# TABLE OF CONTENTS

# LIST OF TABLES

LIST OF FIGURES

# Chapter 1

# Introduction and Background

## 1.1 Introduction

Included in the planning of many engineering projects is an environmental assessment. The assessment should determine the immediate and long term effects of the proposed construction project or operation on the surrounding physical, chemical and biological factors that directly influence the ecological community. This includes the determination of biological populations. An estimate of the abundance of any species is essential in the conservation of that species. This is particularly true for the marine environment since so little is known about this region and the life living within it.

The ocean still remains an alien and unexplored territory. Study of those ocean creatures most closely related to humans, the cetaceans or marine mammals, can provide insight into the ocean world. A decrease in a particular whale population may indicate a serious problem with pollution, a decline of food sources (for both cetaceans and man) or climatic changes. Identification of these problems can be made by noting a significant drop or increase in the abundance of cetacean populations.

Determining the population of marine mammals using methods

developed for land is not practical. Visual observations of whales are difficult since most spend a majority of their time underwater, far from land. Sperm whales (Physeter macrocephalus) for instance, can spend over an hour underwater at depths much deeper than 1000 meters (Watkins and Moore, 1982). They come to the surface to breathe for only brief periods and are not easily spotted even at the surface. Sperm whales make distinctively loud clicking noises underwater that can easily be heard up to nine kilometers away. Other cetaceans are also known to vocalize underwater. Therefore, rather than using visual sampling techniques, it may be possible to acoustically estimate the population density of cetaceans using a three-dimensional version of the line transect theory, a method commonly used to determine biological populations on land.

## 1.2 Objectives

The objectives of this study are to (1) develop a computer program that will simulate two and three dimensional line sampling methods that are used to determine population densities in the marine environment, (2) to verify the simulation program and (3) develop hypothetical test cases and examine the results generated by the simulation program.

## 1.3 Background

Prior to designing a computer simulation it is important to examine and understand the environment to be modeled. The computer program developed and explained in this report is designed to simulate the underwater acoustic detection of sperm whales swimming in the ocean environment using line transect sampling methods. Therefore, an understanding of sperm whale behavior, underwater acoustics and line transect theory are necessary to fully understand the program. Guidelines for the program were set based on these three factors.

Line transect theory has been used since the early 1930s to estimate the population densities of wildlife (Burnham et al., 1980) The study of underwater acoustics, dates back to Leonardo da Vinci, although serious research of the subject did not begin until the early twentieth century. Antisubmarine warfare in the World Wars and, to some extent, a need for better navigational equipment, were the catalyst for recent underwater acoustic research. Today underwater acoustics is used not only in submarines and navigation, but also for fish detection, underwater surveying, seismic exploration, oceanography and bioacoustics (Haines, 1974). Biologist have discovered that underwater acoustics provides a method to explore life in the ocean that would otherwise be impossible to study. The ability of whales to communicate and echolocate using an internal sonar system was discovered by and is studied using underwater acoustics (Harrison, 1988).

Whales belong to the animal order cetacean, mostly aquatic marine mammals including whales, dolphins, porpoises and related forms with a large head, fish-like nearly hairless body and paddle-shaped forelimbs. Sperm whales, made famous by Herman Melville in his novel "Moby Dick", have special features and behaviors that make them unique in comparison to other whales. They rank as one of the largest of all cetaceans and are the largest of the toothed whales. Their physical features include a large head that projects well beyond the tip of a narrow lower jaw. In front of the head and above the upper jaw there is a spermaceti organ containing a special wax-like substance. It is believed that this waxy substance plays a role in adjusting the sperm whale's buoyancy during the changing pressures encountered in deep dives. Sperm whales dive deeper and stay down longer than all other whales. Their vocalizations are also unique in the ocean world (Harrison, 1988).

One characteristic feature of the sperm whale is its large size. Females grow to an average of 13 meters in length and weigh between 6 to 8 metric tons. The males are nearly twice as big as females, averaging 18 meters in length and 15 to 20 metric tons. Lone bulls reaching 24 to 30 meters in length and 27 to 29 metric tons have been sighted (Cousteau, 1986).

The principal habitat of sperm whales appears to be between 40 degrees north and 40 degrees south latitude (Cousteau, 1972).

Large, older males may be found in polar waters where they feed on giant squid. They usually travel alone or in pairs. Younger males may form groups of up to fifty whales and stay in lower latitudes, but as they grow older their group size decreases and the hunting range increases. Adult females, calves and juveniles are normally found in warmer waters in groups of two to fifty individuals. (Harrison, 1988).

Sperm whales eat just about anything including giant crustaceans, seals, crabs, rays, sponges, jellyfish, dolphins and even sharks. However, sperm whales prefer to eat squid over anything else and the best place to find squid is in very deep water (Cousteau, 1972). It is estimated that a mature sperm whale consumes 3 percent of its weight daily; for a 50 ton male this would mean 1 and 1/2 metric tons of squid each day (Cousteau, 1986). Therefore sperm whales spend most of their time underwater, diving at depths of over 2000 meters where they are known to stay submerged for well over an hour (Watkins, et al. 1985)

Although sperm whales are usually seen in large groups at or near the surface, they tend to disperse out both horizontally and vertically while diving underwater. The whales evidently communicate their locations to each other while underwater because after a dive they move to the same location and depth, and surface within a few meters from each other. Underwater acoustic studies indicate that sperm whales do communicate by making clicking noises

while they are diving (Watkins and Schevill, 1977a).

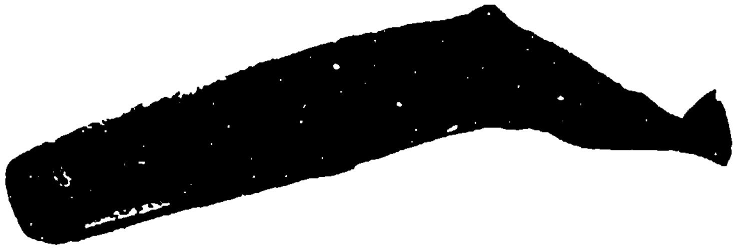
Researchers have found that only a small percentage of a whale group will be seen at the surface at a time. Although only a few are seen at the surface, many more whales can be heard underwater. In fact whales at the surface tend to dive toward the sounds of other distant whales already deep below the surface (Watkins and Schevill, 1975).

While at the surface sperm whales must not need to locate each other acoustically since they rarely make their clicking noises at the surface. They usually begin clicking at the beginning of a dive when a depth of about 5 meters is reached. After this the whales click and can be tracked acoustically by underwater hydrophone.

Sperm whale sounds are entirely impulsive with only clicks. There are no squeals, moans or whistles typical of other cetaceans. Sperm whales can control the level and intensity of their clicks and use a wide range of click rates, from less than 1 per second to more than 75 per second. The relative power of individual clicks can be as high as 75 or 80 decibels relative to 1 dyne per $cm^2$ at 1 meter. The clicks are broad bandwidth pulses with frequencies exceeding 20 kHz, but most are in the frequency range of 2 to 6 kHz. (Watkins and Schevill, 1977b).

Since sperm whales spend so much of their time underwater determining their population in a particular area is extremely difficult. One method called line transect theory commonly used to count animal populations on land has been used to count cetaceans. The theory is applicable to two dimensions and only animals visible on the surface are counted (Cooke, 1985). This eliminates all animals below the surface and in the case of sperm whales this could be a significant part of the population. Sperm whales are ideal subjects for acoustical detection since their vocalizations are so distinctive. If line transect theory can be modified or adapted to three-dimensions it could be used to estimate sperm whale populations based on acoustical detection of animals swimming below the ocean surface.

Currently a population study of sperm whales and other cetaceans is in progress in the northern Gulf of Mexico. A linear hydrophone array is being used to acoustically detect sperm whales. The data obtained during the study could be used in a modified version of the two-dimensional line transect theory for three-dimensions to estimate the current population of sperm whales in the northern Gulf of Mexico. Once the population is known, a baseline may be set to use as a gage to measure the impact on the sperm whales of continued or increased industrialization of the offshore regions. Figure 1.1 shows the distribution of sperm whale sightings and captures in the Gulf of Mexico.



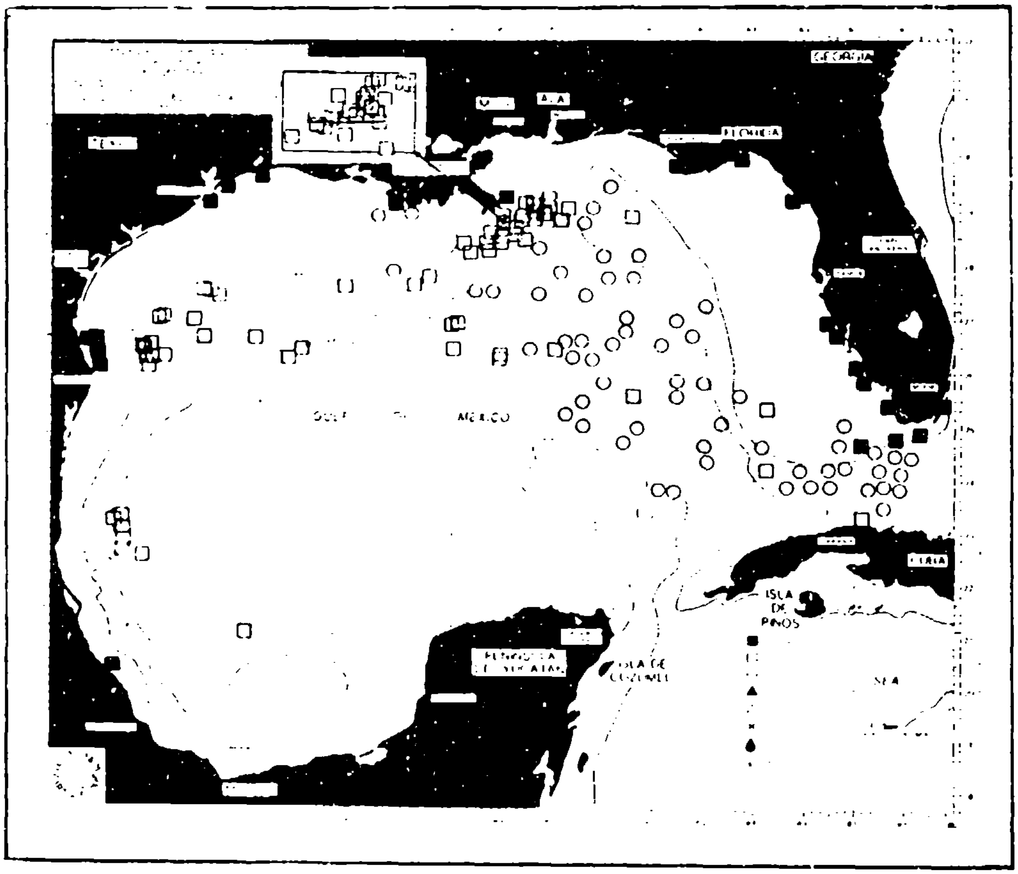



FIGURE 1.2 - Sperm whale (above) and sperm whale sightings in the Gulf of Mexico. Recent surveys in the Gulf have resulted in many sightings of sperm whales from widely scattered locations. Sperm whales are found in deep oceanic water and along continental slopes (Jefferson et al., 1992).

# Chapter 2

# Underwater Sound

## 2.1 Basics of Underwater Sound

Sound is caused by progressive longitudinal pressure waves within an elastic medium such as air, water or solid rock. The speed of sound waves are dependent on the density and compressibility of the medium in which they are traveling. Since sound waves are three dimensional they propagate outward in all directions from the energy source. The existence of one sound wave does not affect the existence or properties of another sound wave, even if they both occupy the same space at the same time. However, sound waves do interfere with each other both destructively and constructively. Sound waves can be reflected, refracted and diffracted (Berg and Stork, 1982).

The propagation of sound in an elastic medium can be described mathematically by solutions of the wave equation using appropriate boundary conditions. The wave equation is a partial differential equation relating the acoustic pressure, p, to a location in space commonly represented by x,y,z, coordinates relative to an origin and time, t. The equation may be expressed as:

$$\frac{\partial p^2}{\partial t^2} = c^2 \left[ \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} + \frac{\partial^2 p}{\partial z^2} \right] \qquad (2.1)$$

where **c** is the velocity of the sound waves traveling through the medium (Urick, 1983).

The speed of sound traveling in the ocean is dependent on the properties of sea water. Sea water is not homogeneous. There are variations in temperature, salinity and pressure. Since these properties vary greatly depending on the water depth, geographic location, season and time of day, the speed of sound in the ocean also varies greatly. The sound speed in the ocean can only be estimated by empirical equations. One equation for the speed of sound in the ocean aptly demonstrates the complicated relationship between sound speed and properties of the medium:

$$C = 1448.96 + 4.591T - 5.304 \times 10^{-2}T^2 + 2.374 \times 10^{-4}T + 1.340(S-35) + 1.630 \times 10^{-2}D + 1.675 \times 10^{-7}D^2 - 1.025 \times 10^{-2}T(S-35) - 7.139 \times 10^{-13}TD^3 \tag{2.2}$$

where D is the depth in meters, S is the salinity in parts per thousand and T is the temperature in degrees Celsius. Limits of this particular formula are $0 < T < 30°$, $30 < S < 40$ 0/00, $0 < D < 8000$ m (Mackenzie, 1981). This formula shows that the speed of sound in the ocean is most dependent on temperature. Only at great depths does the pressure become a significant factor and the salinity adds only a minor contribution to the relationship.

The complicated relationship between the speed of sound and the ocean medium results in many interesting phenomena. The ocean may

be divided into several layers that affect the speed of sound in different ways as can be seen in Figure 2.1. At and just below the surface is the mixed layer. In this layer the density, temperature and salinity of the water are constant as a result of turbulent mixing by the wind and other forces. Here the sound speed increases with increasing depth. The next layer is a combination of the seasonal and main thermocline where the temperature decreases rapidly with depth. Since sound speed is mainly dependent on temperature it too decreases with depth in the thermocline layer until the deep isothermal layer is reached. The temperature in the isothermal layer does not vary much but the pressure does, increasing with depth and causing the sound speed also to increase.

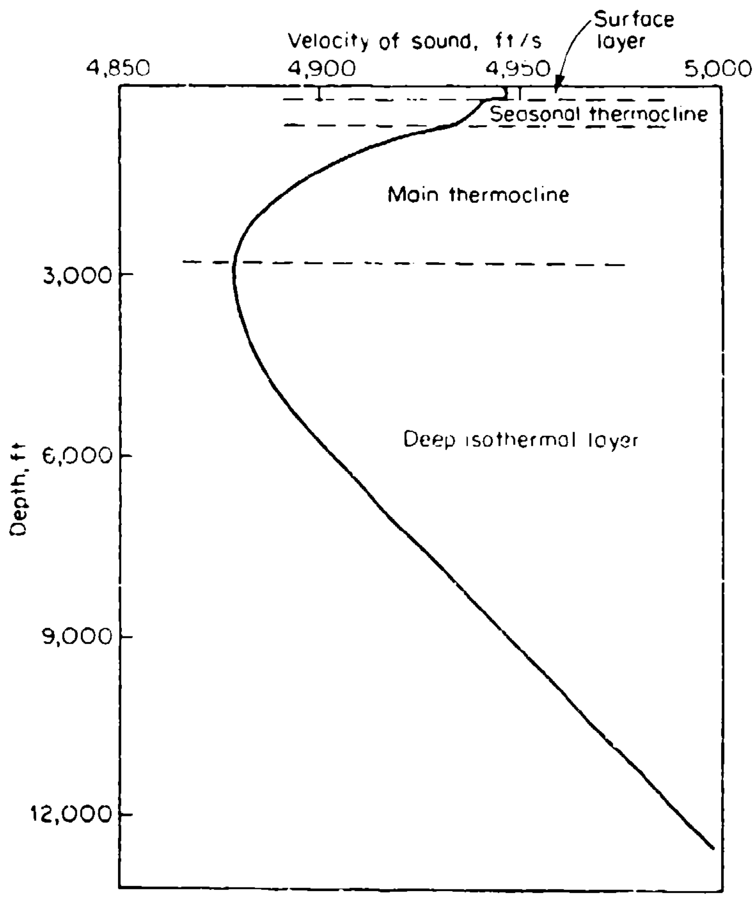



FIGURE 2.1 - Typical deep-sea sound velocity profile (Urick, 1983).

The depth of each layer depends on the geographical location, season of the year and time of day. For instance, during the day the surface layer of the ocean is heated and no longer has a constant temperature or salinity since the layers at the very top are warmed and evaporate more quickly than those below. However, at night the mixed layer becomes more defined as the surface layers cool and are mixed by wave action and the sun is no longer generating a temperature gradient in the water. The seasonal thermocline usually appears during the summer and fall when the water at the surface is warm, but becomes increasingly cooler with depth. In the winter and spring, the difference between surface temperatures and deep layers is minimal so the seasonal thermocline is not as distinguishable from the mixed layer. Correspondingly there is a strong thermocline at lower latitudes where the sun is most prevalent. At high latitudes the surface temperature can be as cold or much colder than the water temperatures below and so there is no thermocline. The variations in sound speed profiles for various locations around the world are compared in Figure 2.2 (Urick, 1983).

Sound waves are refracted and reflected as they travel through layers of different temperatures, densities and/or salinities. They are also reflected at the water surface and at the ocean bottom. A sound ray diagram, Figure 2.3, demonstrates how acoustic rays are bent due to refraction towards regions of lower sound speed and away from regions of high speed. This bending results in

a "shadow zone" where direct sound rays can not penetrate. Shadow zones begin at sound-speed maxima. A minimum in the velocity profile tends to channel sound waves to its own level or depth. This results in a sound channel. Figure 2.4 demonstrates how sound rays are trapped into a channel when the sound source is located at the lower boundary of the main thermocline.

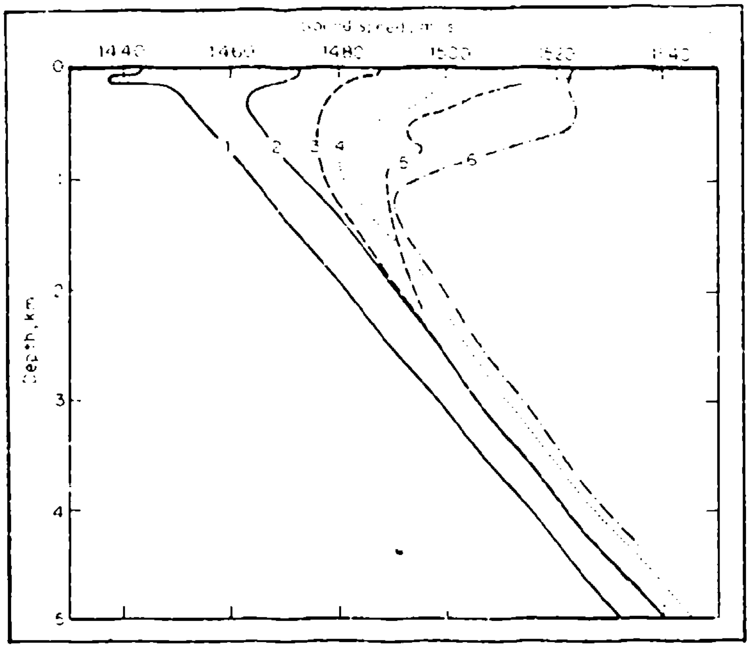


FIGURE 2.2 - Characteristic velocity-depth profiles for the deep-ocean areas of the world. 1) Antarctic Ocean, 2) North Pacific, high latitudes, 3) Southern oceans, high latitudes, 4) Pacific and South Atlantic, low latitudes, 5) Indian Ocean under influence of Red Sea outflow, 6) North Atlantic under influence of Mediterranean Sea outflow ("Ocean Science Program of the U.S. Navy", Office of the Oceanographer of the Navy, Alexandria Virginia, 1970).

Near the surface, in the mixed layer, a sound channel is created by a pressure effect. The pressure increases with depth, increasing the sound speed and therefore bending the sound rays away from the sound speed maximum and back up toward the surface. The sound is

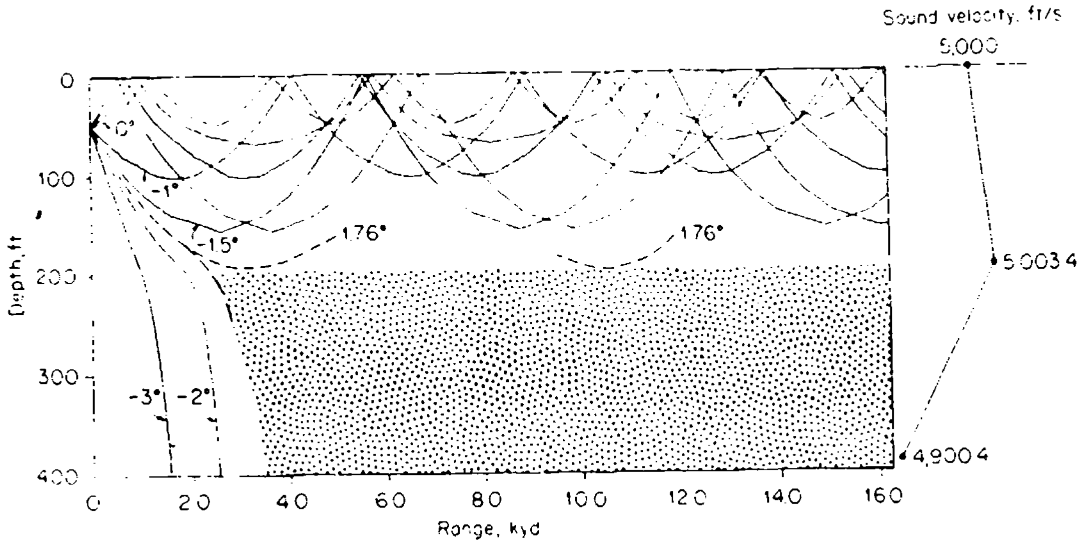



FIGURE 2.3 - Ray diagram for a sound source in a typical mixed layer. Depending on the angle of the ray when leaving the source, rays are either trapped in the layer, bending away from the sound velocity maximum at the bottom of the layer and up to the surface or rapidly exit the layer and travel down to deeper layers. The dotted area represents a "shadow zone" beneath the mixed layer created by the bending of the sound rays. It is called a shadow zone since the sound rays do not directly penetrate into this region and therefore an acoustic "shadow" is created (Urick, 1983).

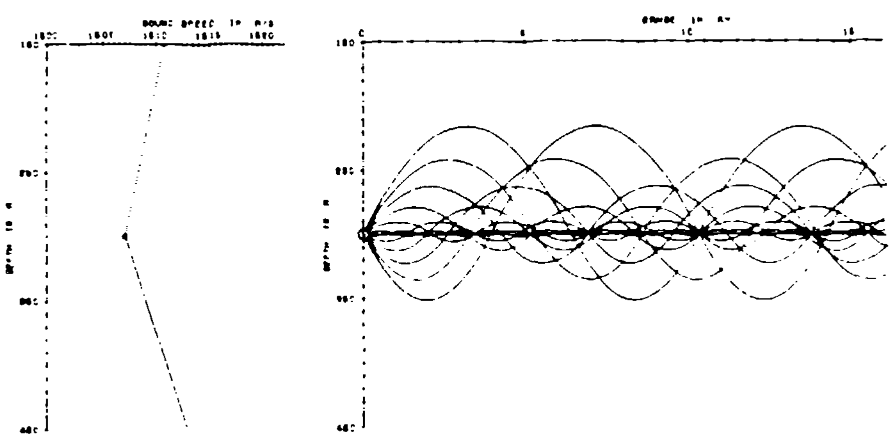


FIGURE 2.4 - Diagram showing how sound rays are trapped in a channel at the bottom of the main thermocline, a point of minimum sound velocity as shown on the accompanying velocity profile. The sound trapped in this channel at very low frequencies is subject only to spreading and very little attenuation loss (Coates, 1989).

trapped in the layer and may propagate long distances by successive reflections from the sea surface and subsequent bending away and back upward from the depth where the sound speed is a maximum.

A minimum in the sound speed exists at the bottom of the main thermocline where the temperature becomes constant but the pressure continues to increase with increasing depth. This minimum in the sound speed is the axis of another sound channel known as the deep sound channel or SOFAR (Sound Fixing and Ranging). The deep sound channel occurs at depths of 800-1200 meters in low latitudes and can be found near or at the surface in high latitudes depending on the depth of the thermocline.

In regions of very high latitude there is no thermocline. The temperature of the water is nearly the same at all depths. In fact, it may be even colder at the surface then at deeper depths. The sound speed simply increases linearly with increasing depth or increasing pressure since temperature effects are minimal (Urick, 1983).

## 2.2 Spreading and Attenuation

As sound travels through the ocean it will, with distance and time, weaken and become distorted until it is no longer distinguishable. This weakening and distortion is due to a variety physical properties of the ocean environment.

The sound wave intensity decreases as the distance from the source increases because of spreading loss and attenuation. Sound waves are also absorbed, reflected, refracted and diffracted by various boundaries, sound channels and the sea water itself.

Spreading loss is not caused by the sound medium. Like the circular ripples spreading outward from a raindrop in a puddle, sound waves spread equally in all directions from their source. If there is no energy losses to the medium then the power generated by the source should be the same at equal distances in any direction around the source. This can be represented by an infinite number of spherical shells or surfaces of the same power around the source, see Figure 2.5.

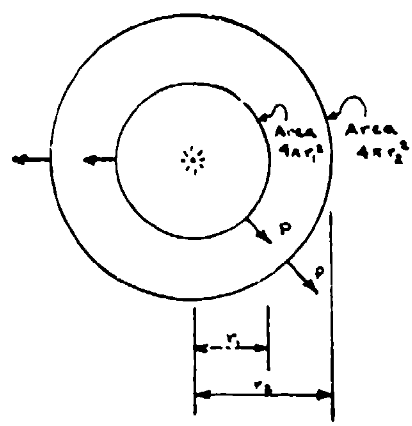



FIGURE 2.5 - Spherical spreading.

Power, $P_w$ is the sound intensity multiplied by the area, so for a sphere

$$P_w = 4\pi r_2 I_1 = 4\pi r_2 I_2 = \dots \tag{2.3}$$

where $I_1$ is the intensity at a distance $r_1$ from the source and $I_2$ is the intensity at a distance $r_2$ from the source. The power remains

constant with distance from the source but it is spread over an ever increasing surface area resulting in a decrease in intensity or a loss of the strength of the sound that is proportional to the inverse of the square of the distance from the source (Urick, 1983).

Attenuation is the scattering and absorption of sound wave energy. Scattering is caused by the reflection, diffraction and reradiation of sound by macroscopic and microscopic inhomogeneities in the medium such as schools of fish or soil particles. Absorption results from various phenomenon including thermal conductivity, viscosity, structural and chemical relaxations and resonant absorption (Caruther, 1977).

## 2.3 Noise

The ocean is a very noisy place. Ambient noise is the term used to describe background noise or any sound not of particular interest to the listener. Figure 2.6 shows some of the many sources of underwater noise. How noisy the ocean is depends on the frequency range of interest and the depth. A pressure spectrum in decibels is used to describe or compare the level of sound of the various sources of noise. Figure 2.7 shows a composite of spectrum levels versus the frequency of various sources of ambient noise.

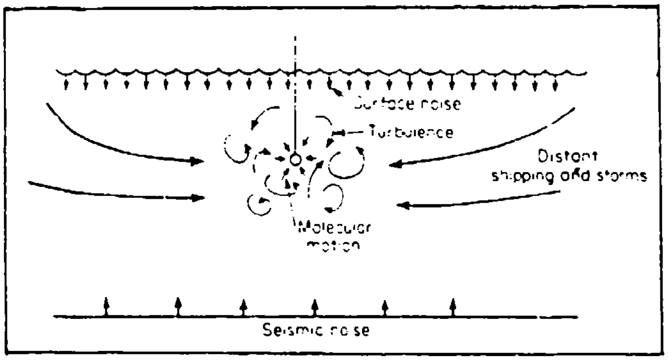



FIGURE 2.6 - Some of the sources of deep-water ambient noise (Urick, 1983).

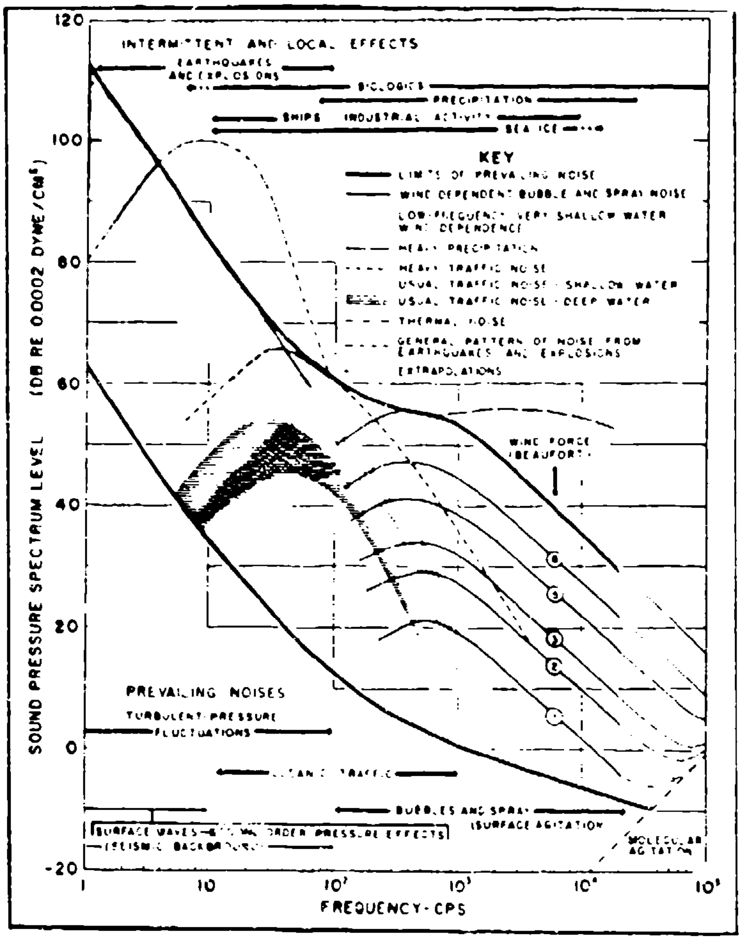



FIGURE 2.7 - A composite of ambient noise spectra, summarizing results and conclusions concerning spectrum shape and level and probable sources and mechanisms of the ambient noise in various parts of the spectrum between 1 Hz and 10 kHz (Wenz, 1963).

Underwater noise may be divided into three main categories: water motion, marine life, and manmade sources (Wenz, 1963).

Water noise includes sounds made by wind on the sea surface, surface waves, internal waves, breaking waves, the impact of rain and spray, and the movement of bottom material by earthquakes and volcanoes. Water noise contributes to the entire under water frequency range well below and far above the frequencies of human hearing.

Continuous movement of the earth's crust has been determined to cause noise at frequencies below 1 Hz. Measurements have shown that other seismic activity including earthquakes and volcanoes add to noise levels from 10 to 100 Hz. Water turbulence that causes varying dynamic pressures produces noise of frequencies ranging from 1-10 Hz. Wind agitation at the sea surface adds to noise in the ocean between the range of 500 Hz to 20 kHz (Wenz, 1963).

Studies of rain noise show that depending on the wind speed and intensity of the rain, the noise created is between 100 Hz to over 10kHz. This kind of noise is be most noticeable in shallow depths of less than 250 meters (Urick, 1983).

Noise from marine life is about as varied as the variety of life found in the ocean. The sounds of biological organisms in the ocean have been studied extensively. The frequency range spans 100

to 10,000 Hz (Wenz, 1963). Examples of some of the more prominent sounds near coastal areas come from snapping shrimp opening and closing their enlarged claws. The sound is similar to the sizzle of frying fat. Croakers, a variety of drumfish, produce a series of taps by the contraction of drumming muscles attached to their air bladder (Knudsen et al., 1948).

Dolphins produce a variety of sounds and baleen whales are known for their singing, particularly the humpback whale. Of particular interest in this study are the unique clicking noises produced by sperm whales (Harrison, 1988). Many varieties of fish make noises mostly when eating. If they are not known to produce a noise themselves, they create noise when digging in gravel or scratching around looking for food. Even barnacles make noise with an occasional click of low intensity (Knudsen et al., 1948).

Noises caused by marine mammals are significant because they are caused by such a wide variety and range of sources. They can fluctuate from hour to hour, day to day or month to month forming an erratic random contribution to the ambient background of the sea.

Noise from ships and other man-made sources is most prevalent near coasts and shipping lanes. Traffic noise or noise from distant shipping has been found to be a significant contributor to underwater ambient noise and often dominates the frequency range

between 20 and 500 Hz (Wenz, 1963). Even in deep water away from the continental shelf low-frequency ambient noise is often dominated by coastal shipping. Sounds generated near the coast propagate seaward where the relatively steep continental slope channels the sound and enhances it like a megaphone used by a cheerleader to shout at a football game. This is demonstrated in Figure 2.8. The sounds are then caught in the deep sound channel and may travel thousands of miles (Urick, 1983).

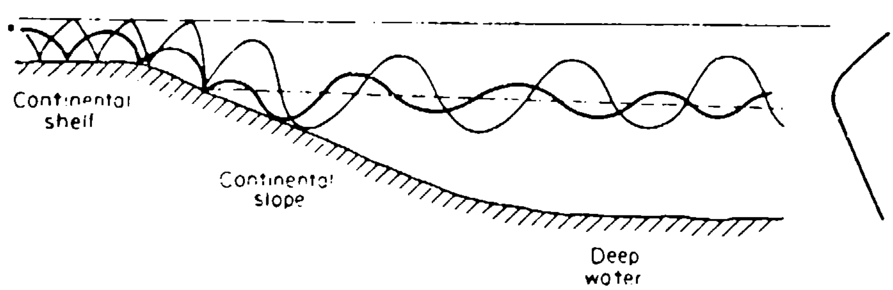



FIGURE 2.8 - Ray diagram showing how coastal shipping noise can propagate to long ranges in deep water in the deep sound channel. The sound velocity profile is shown at the right. Rays originally refracted downward become refracted upward once they reach deep water and the channel axis (Urick, 1983).

Other sources of man-made noise include industrial activity onshore and offshore such as pile driving or drilling. Noise from explosions is similar to that of earthquakes. An explosion produces a wide range of frequencies at close range, but at a distance only lower frequencies are heard (Wenz, 1963). As the activities of humans increase artificial background noise in the

ocean environment will undoubtedly become more prevalent and a nuisance to those trying to listen.

## 2.4 Sonar and the Sonar Equations

The first practical use of man-made underwater sound was to aid in navigation of surface ships. With the use of submarines came the need to navigate completely submerged with no visibility and to detect other submarines, particularly enemy submarines. World War I and II spurred development of underwater acoustics for uses in echo sounding, sound ranging and seismic prospecting. Sonar (sound, navigation and ranging) became the underwater counterpart to radar (Haines, 1974).

In order to design and operate sonar equipment all the complexities and diversities of the ocean environment must be considered. This is done by identifying and quantifying sonar parameters into relatively simple equations. The sonar equations are based on the relationship between the desired and undesired portions of a received signal.

An underwater receiver receives all sound energy, both the desired sounds and the undesired sounds such as snapping shrimp or the implosions of tiny air bubbles around ship propellers. The desired portion of the received sound energy is called the signal while the remainder is called the background. The objective in designing

sonar equipment is to increase the response of the sonar system to the signal and decrease the response to background noise or in other words increase the signal to background noise ratio. If the level of background noise is higher than the signal level, the signal is not be detected. Therefore, the sonar equipment must be designed so that the signal level is equal to or greater than the background noise.

There are two types of sonar equations, active and passive. Active sonar requires the observer to send a signal that is reflected by an object of interest or target and then received by the observer. The reflected signal provides the listener with information about the location of the target, size and shape. Active sonar is used in anti-submarine warfare, navigation, fish detection, surveying and shipboard positioning systems (Haines, 1974). Passive sonar relies on the target itself to make a noise, and the observer plays no active part in generating signals. All the observer has to do is receive. Passive sonar is also used in antisubmarine warfare and in bioacoustics (Stefanick, 1987).

There are three significant parameters which need to be considered when designing sonar equipment and these are shown in Table 2.1. All parameter elements are expressed in decibels, dB, relative to a unit pressure, such as dyn/cm$^2$ or μPa.

Table 2.1. The three parameters in designing sonar equipment (Urick,1983).

| | |
|---|---|
| Parameters determined by equipment: | projector source level, SL<br>self noise level, NL<br>receiving directivity index, DI<br>detection threshold, DT |
| Parameters determined by the medium: | transmission loss, TL<br>reverberation level, RL<br>ambient noise level, NL |
| Parameters determined by the target: | target strength, TS<br>Target source level, SL |

The projector source level, SL, is a measure of power flux delivered into the water by a source at a standard range from the presumed acoustic center of the source. Self noise, NL, is that unwanted noise received along with the desired signal that is caused by the acoustic equipment itself or the ship on which the equipment is located (Coates, 1989).

The equipment used to receive the incoming signal does not necessarily receive sound equally from all directions. Only receivers that are perfectly spherical and respond equally in all directions and that are in a uniform medium of infinite extent can be considered omnidirectional. Since the ocean is certainly not an infinitely uniform medium and equipment is never perfect, the array gain or the directivity index, DI, is included in the sonar parameters to account for the directivity of the acoustic system.

The detection threshold, DT, is defined as the ratio of signal power to noise power. When a signal is received it has to be distinguished, either by the equipment or by a human observer, above all the other sounds being received. If the threshold is set too low, the observer or equipment may identify signals as being the one desired when it really is not. If the threshold is set too high, only very strong signals will be detected since those will be the only ones allowed through the equipment from the receiver.

Transmission loss, TL, includes all possible areas of energy loss into the surrounding medium between the signal source and the receiver. Transmission loss includes spreading and attenuation in the medium.

The reverberation level, RL, is used only in active sonar calculations. If a signal is generated to detect a target, it is reflected off and scattered by not only the target but every other object in the area including bubbles, suspended particles, fish, the ocean surface and the seafloor. The reverberation level often is the primary limitation on an active sonar system so it must be estimated and included in the sonar design process.

The ambient noise level, NL, is the level of the undesired background noise also being received in addition to the desired signal. The ambient noise level may be so great that the desired signal is not detectable.

The target strength, TS, refers to the intensity of sound returned or reflected back by the target at a standard distance from the acoustic center of the target. The source level is similar to the target strength except it is the intensity of the radiated sound produced by the target itself at a standard distance (1 meter) from the acoustic center of the target.

The derivation of the active sonar equation is quite simple even though the sonar parameters can be extremely complex. The equation may be derived by describing a signal emitted from an acoustic system into the surrounding environment with a specific source level, SL. Before the signal can reach the target there is transmission loss, TL, which decreases the source level. The signal then is returned by the target that has a given target strength, TS. The returned signal also looses intensity due to transmission loss, TL. There will be a background noise level, NL, received with the returned signal that is partly reduced by the equipment directivity index, DI. When the returned signal minus the transmission loss and background noise is just detectable it should equal the detection threshold. This can be expressed as the active sonar equation:

$$SL - 2TL + TS - NL + DI = DT \qquad (2.4)$$

The passive sonar equation is less complicated than the active sonar equation. In this case the source level is determined by the

target. The transmission loss is only between the source and the listening receiver. The noise level is the same since it is dependent on the medium. The directivity of the acoustic system is still included. The detection threshold is reached just as the signal less the transmission and background noise becomes detectable. The equation for this relationship can be written:

$$SL - TL - NL + DI = DT \tag{2.5}$$

These sonar equations are only generalizations. Each sonar parameter will fluctuate with time. In addition, the ocean is an inhomogeneous medium with irregular boundaries and also in constant motion. Even the properties of the acoustic equipment system will vary, so there are no constants. However, by using the sonar equation, a good estimate can be obtained for designing and operating a sonar system (Urick, 1983).

## 2.5 Underwater Acoustics Instrumentation

Underwater acoustics equipment must be designed to withstand the corrosive environment and high pressure in the ocean. The human ear is not designed to hear sounds underwater, but is certainly ideal for hearing sounds in air. Therefore, underwater acoustic equipment must be able to transform the sounds generated underwater into sounds audible in the air or into a visual interpretation. An active sonar also requires that sound be generated underwater. The

equipment used underwater must be able to match the acoustical impedance of water which is about 4000 times greater than that in the air (Caruthers, 1977).

A transducer is used to accomplish the task. A transducer converts mechanical, chemical or electrical energy into acoustic energy in the water medium or it converts acoustic energy present in the water medium into electrical energy. A transducer used to convert other forms of energy into acoustic energy is called a sound projector. A transducer used to convert acoustic energy into electrical energy is called a hydrophone.

Underwater transducers are most commonly made from materials that possess special properties, either electrostriction or magnetostriction, that convert energy from one form to another form. Magnetostriction is used in the design of some types of low frequency transducers, but electrostriction is most common. Quartz is a well-known natural piezoelectric material, but the most versatile and most frequently used materials are ceramics made of lead zirconate titanate (PZT). This ceramic is produced as a powder that is compressed and fired into various shapes such as rings, tubes, discs or plates. They may be cut or ground to produce special acoustical properties (Coates, 1989).

Hydrophones are specifically used to receive the pressure variations of sound waves in the ocean. Figure 2.9 shows the plans

for four different types of hydrophones. A hydrophone usually consists of piezoelectric material, usually ceramic, with electrical leads connected to transfer voltage generated by sound waves to the surface. The voltages generated are very small, on the order of microvolts, with high impedance, and therefore need to be greatly amplified to be useable. A preamplifier located near the hydrophone amplifies the voltage signal and lowers the impedance (the ratio of the pressure to the volume displacement at a given surface in a sound-transmitting medium). It is important to lower the impedance before the signal is sent to the surface. A high impedance signal will collect additional noise from the hydrophone, the wires going to the surface and at the receiving equipment. A transformer may also be used to lower the impedance.

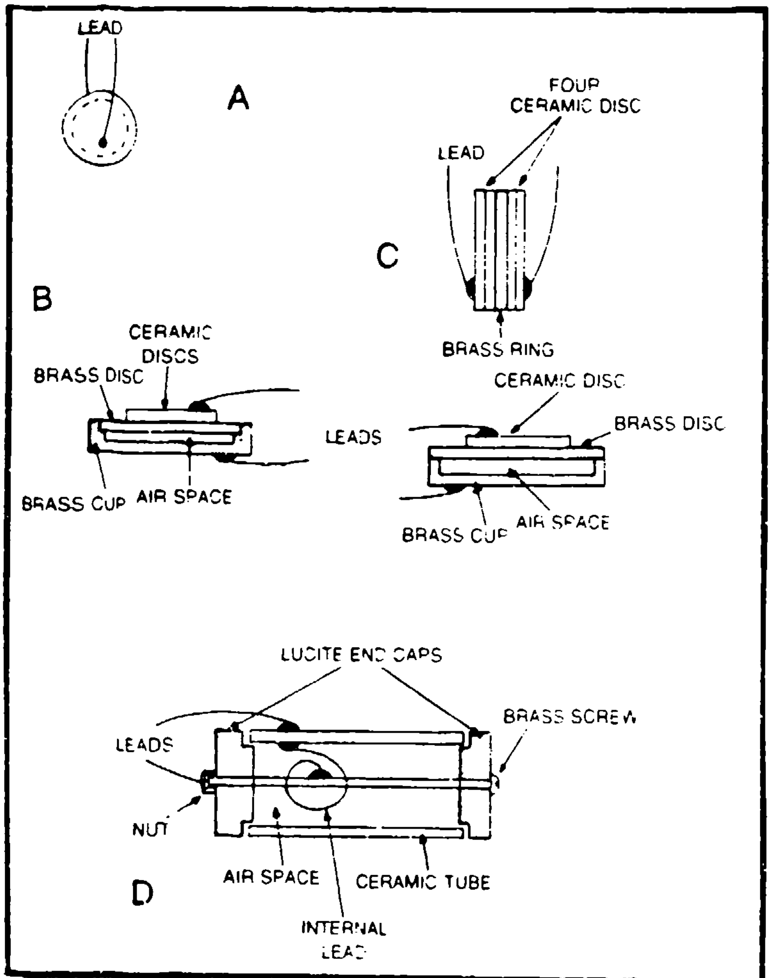



FIGURE 2.9 - Four different hydrophone ceramic transducers (Watlington, 1979).

The low impedance signal is transferred from the hydrophone to the surface via a cable. At the surface the signal must be amplified again because it will still be too weak to use. A power amplifier is employed to increase the amplification of the signal so that it is strong enough to drive headphones, a speaker, or other receiving device such as an oscilloscope (Watlington, 1979).

Single hydrophones are used only for special situations such as research or measurement work. Hydrophones are usually combined to form one unit called a hydrophone or transducer array, a hydrophone array being used exclusively as a receiver of underwater sound.

Using an array of hydrophones provides several benefits. The array is more sensitive than a single transducer because a group of hydrophone elements generates more voltage when connected in series or more current if connected in parallel than a single element exposed to the same sound wave. An array of hydrophones has directional properties enabling it to determine the direction of individual sounds. The array also has an improved signal-to-noise ratio compared to a single hydrophone since it can be directed for maximum reception of an desired signal (Urick, 1983).

The sensitivity of a hydrophone is the number of volts developed across the electrical terminals per unit sound pressure. A hydrophone is omnidirectional if the sensitivity is the same for a sound incident from all directions. If a hydrophone has more

sensitivity in one direction than in any other it is a directional hydrophone. The direction of maximum sensitivity is called the acoustic axis.

The receiving response of a transducer is represented as a function of spherical angles about the transducer relative to the acoustic axis. Figure 2.10 is a three-dimensional representation of the response of a continuous linear hydrophone array of length L with half the beam angle equal to 25λ/L, where λ is the wavelength. The acoustic axis is perpendicular to the line of the array.

A transducer beam pattern is a two-dimensional graphical plot of the signal of a projector or the sensitivity of a hydrophone as a function of an angle measured from the acoustic axis. The plot can

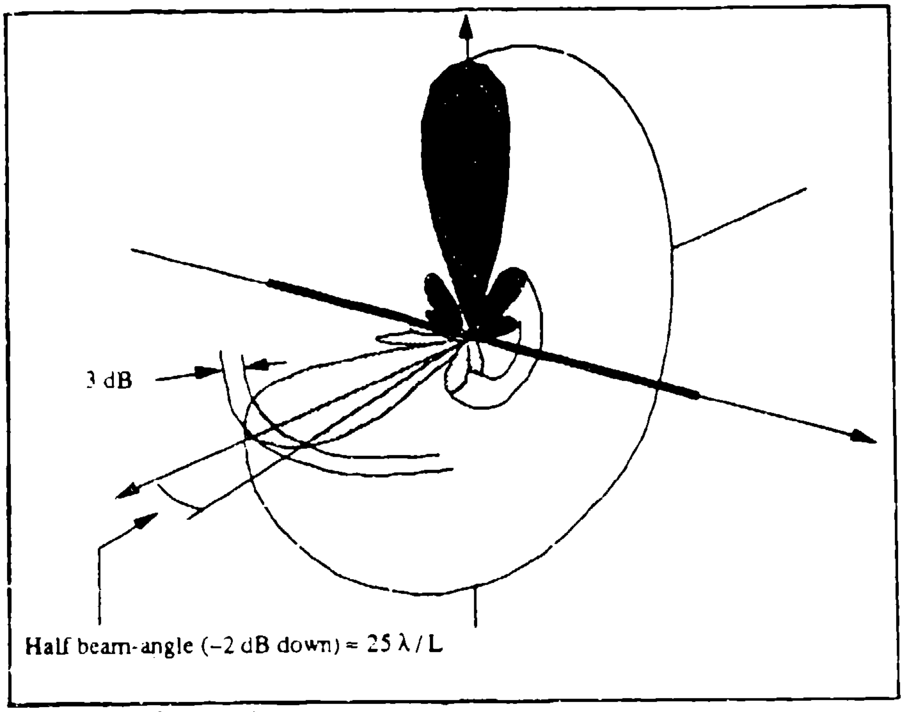



FIGURE 2.10 - Directional response of a continuous line array of length L (Coates, 1987).

be either in rectangular or polar coordinates. Usually the pressure level or relative sensitivity is in decibels, dB. Plots of the sensitivity show where the major beam is located as well as all secondary beams, called side lobes. The sensitivity of the side lobes may be decreased, but this results in an increase in the width of the major beam (Albers, 1969).

Hydrophone arrays may be of various configurations such as rectangular, circular, or even spherical in formation. The simplest configuration is a linear array of equally spaced, uniform hydrophones. Figures 2.11 and 2.12 show various beam patterns obtained from a linear array of uniformly spaced hydrophones. Figure 2.11 shows that as the number of hydrophones increase, the directivity index, DI, increases. Figure 2.12 demonstrates that if the number of hydrophones remains constant, but the spacing is

increased by multiples of the wavelength, the number of lobes increases and the directivity decreases (Coates, 1989).

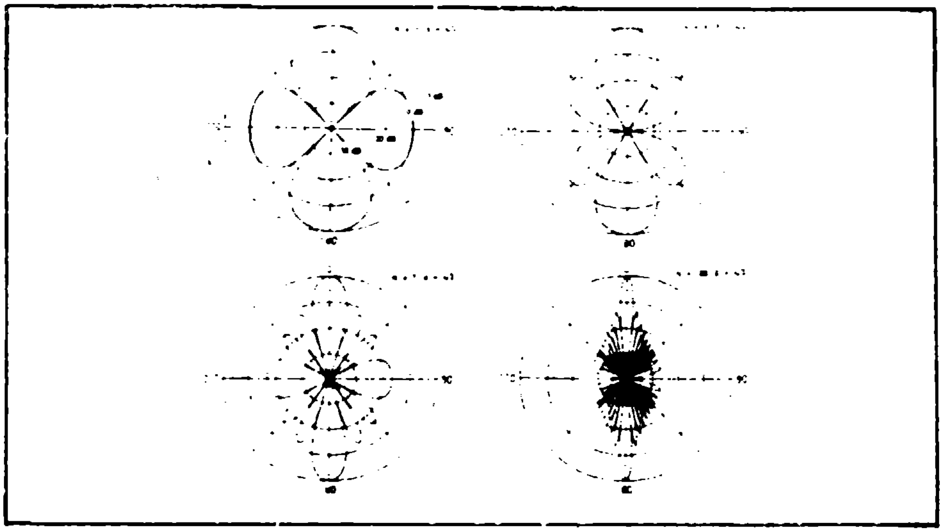


FIGURE 2.11 - Polar transducer beam pattern for a uniformly spaced hydrophone array, with one-half wavelength spacing between hydrophone elements. As the number of hydrophones is increased from one to twenty the beam pattern changes dramatically. The zero angle direction corresponds to broadside sensitivity (Coates, 1987).

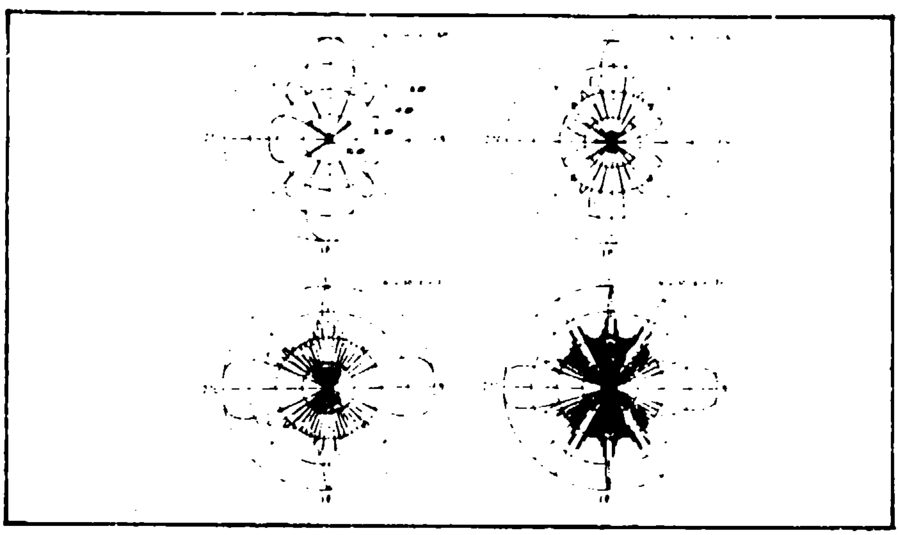


FIGURE 2.12 - Polar transducer beam pattern for a uniformly spaced 10-hydrophone array. The spacing between individual hydrophones is increased to demonstrate the effects on the beam pattern.

## 2.6 Linear Arrays

Currently a ship-towed linear array is being used in the northern Gulf of Mexico to detect sperm whales and other marine life (Gulfset contract). A linear array is used because it is ideal for passive sonar detection. Linear arrays may also be hull mounted or tethered, but a towed linear array can be positioned well astern of the towing ship reduce self noise caused by the ship hull, propulsion and machinery. A broadside acoustic beam may be obtained using the shipboard electronics and this allows assessment of the target location. A linear array may be easily installed and removed from towing vessels and minor repairs can be performed at sea.

The linear array system consists of three essential parts: 1) the wet end, 2) the shipboard handling gear, and 3) the on-board electronics. Towed linear arrays must be designed with sufficient buoyancy to travel horizontally at a given ship speed. A linear towed array configuration is shown in Figure 2.13.

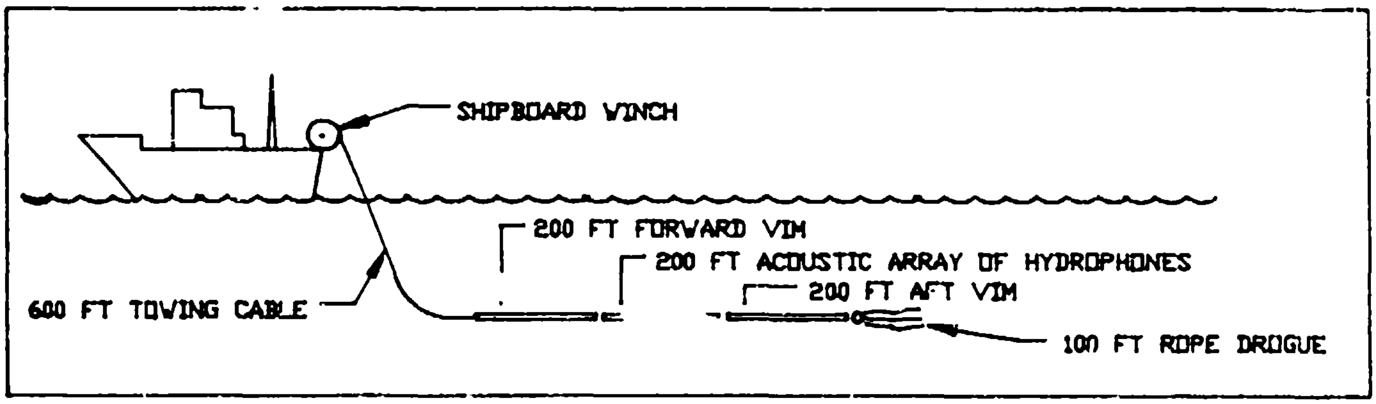



FIGURE 2.13 - Ship-towed linear hydrophone array configuration (Anderson and Evans, 1980).

Figure 2.14 shows a typical hydrophone arrangement with hydrophones grouped for optimum reception of specific sound frequencies. This is done by spacing the hydrophones a set distance from each other within the group and then carefully spacing each hydrophone group. The distance between groups is determined first by finding the center of each desired frequency range and the corresponding wave length. Then the hydrophone group(s) designated to cover this frequency range is aligned in the array to represent that specific wavelength. (Anderson and Evans, 1980).

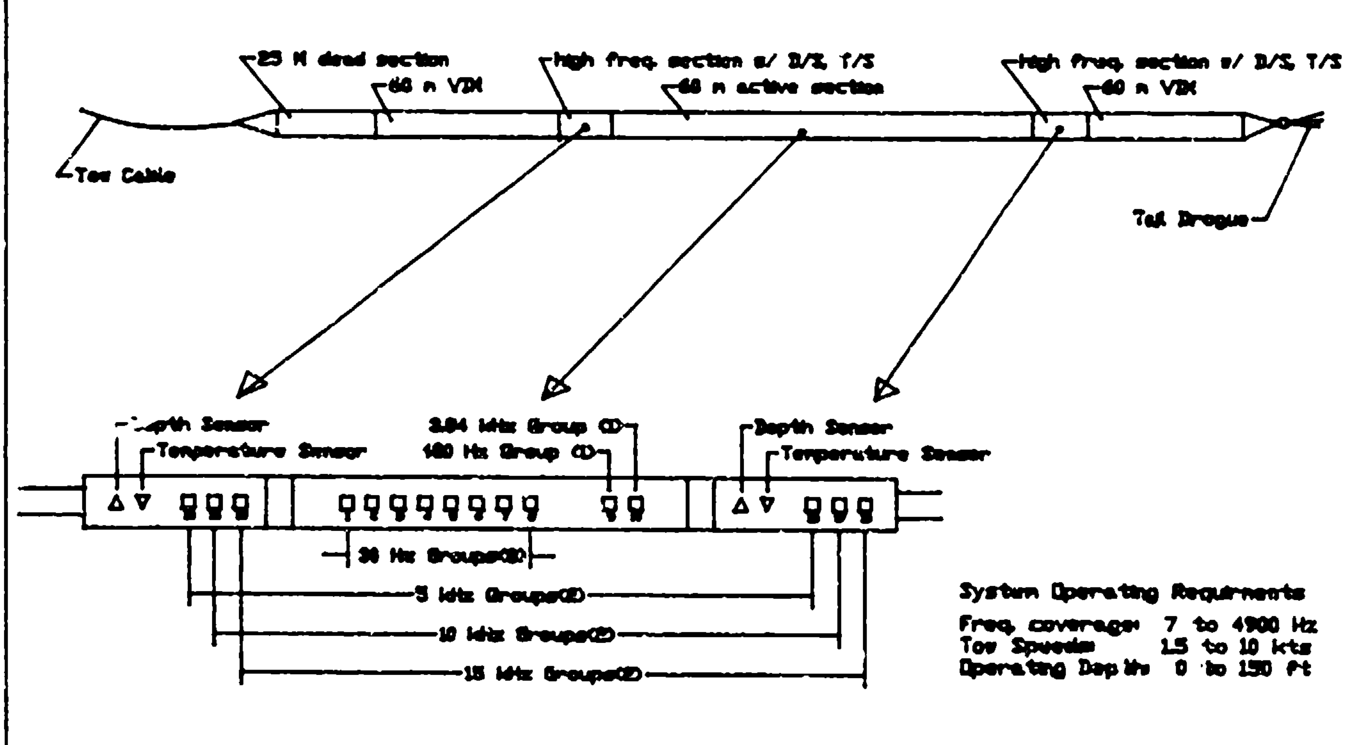
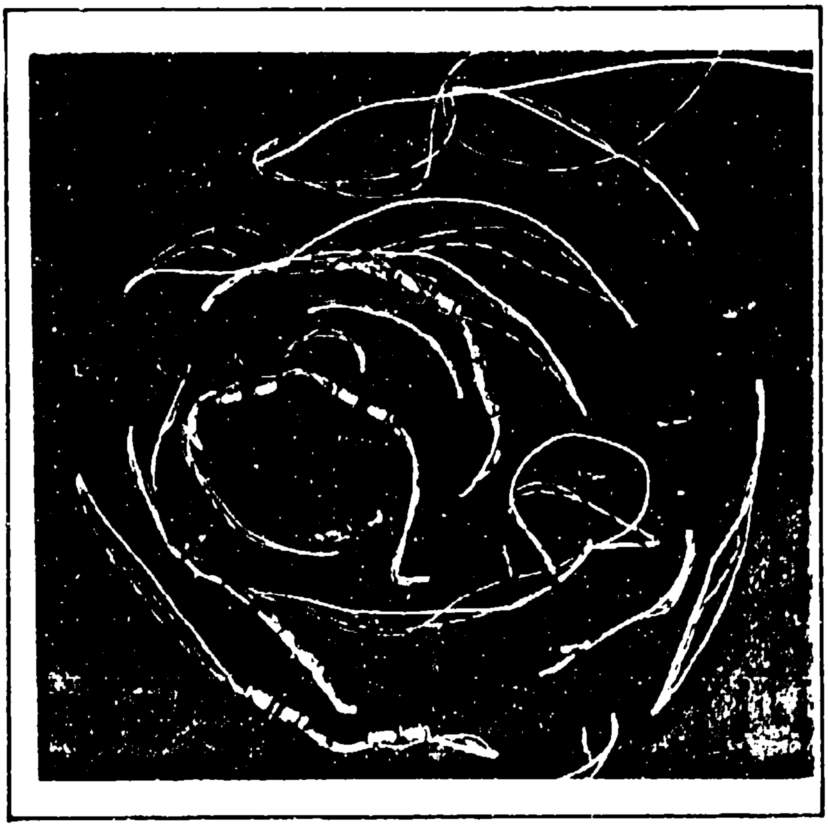



FIGURE 2.14 - Schematic of linear hydrophone array (Anderson and Evans, 1980).

Included with the array of hydrophones are depth and temperature sensors. Figure 2.15 shows the internal wiring and hydrophones of a linear array. The array, sensors and wiring are encased in plastic tubing filled with special oil for waterproofing and

protection. The array is towed by the ship via a cable. In order to reduce the inherent cable motions during towing a Vibration Isolation Module (VIM) is located between the cable and the hydrophone array. At the trailing end of the array a tail swivel and rope drogue provide additional drag to keep the array straight. Another VIM is located between the end of the hydrophone array and the tail swivel to decrease whipping motions as shown in Figure 2.14.

FIGURE 2.15 - Hydrophones and wiring inside a linear array (Watlington, 1979).



The main component of the shipboard handling gear is the winch system for storing, reeling-out and reeling-in the towing cable and the array. A grounded shielded electrical cable connects the tow cable on the winch to the on-board electronics.

The on-board electronics system includes two amplifiers. The conditioning amplifier receives the acoustic (electrical) signals from the array and passes them on to a summing amplifier and spectrum analyzer. The summing amplifier accumulates hydrophone group signals to produce a broadside acoustic beam. The spectrum analyzer provides a visual display of the frequency, time and amplitude of the electrical signals. The visual display is a near-real time image of both the desired signal and the background noise that the array is receiving underwater. A multi-track tape recorder preserves all raw and processed data for later laboratory analysis. Figure 2.16 is a schematic drawing of the on-board system (Anderson and Evans, 1980).

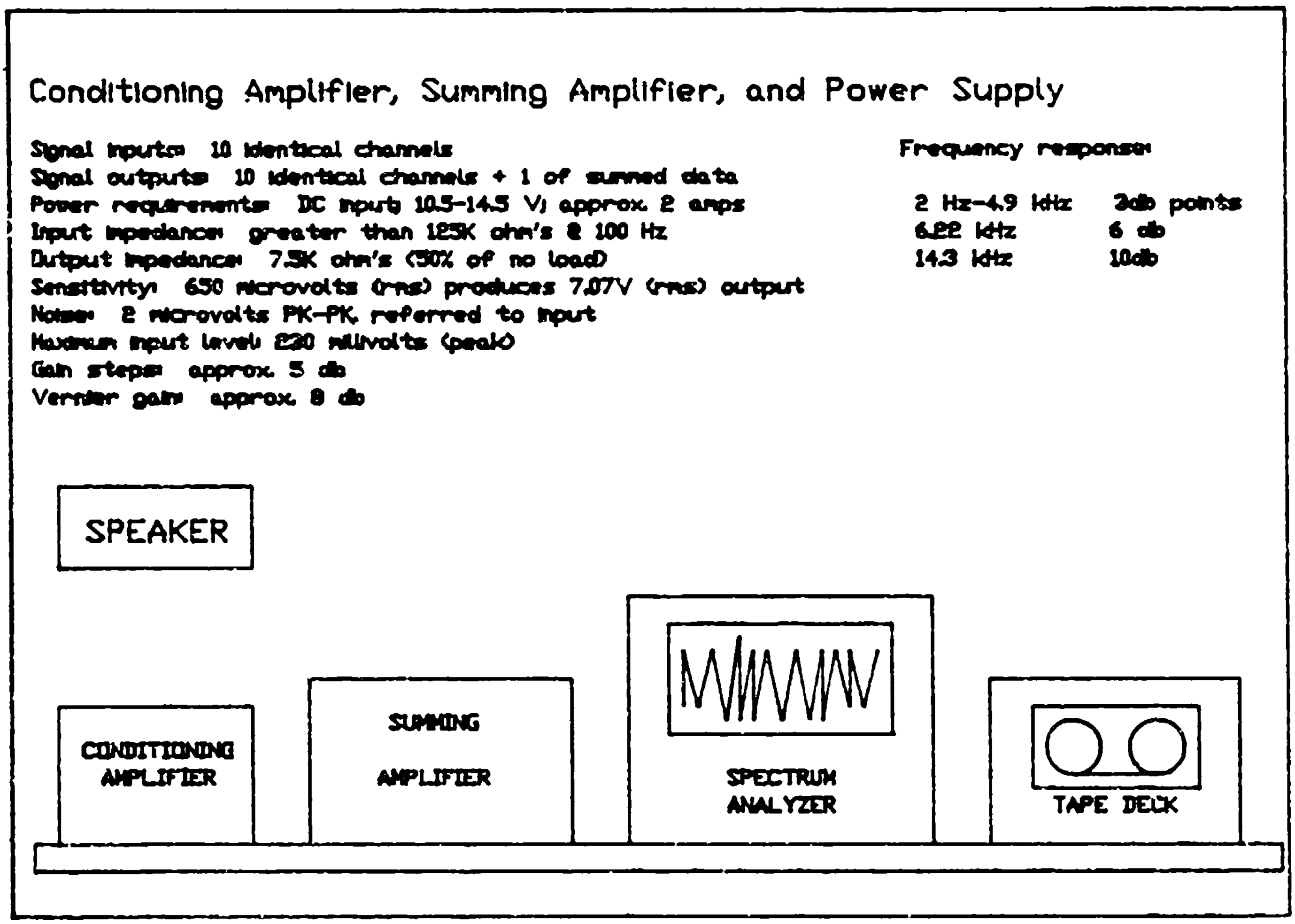



FIGURE 2.16 - Schematic showing on-board acoustic equipment.

# Chapter 3

## Line Transect Theory

The best way to determine the population of a particular animal in an area, or its population density, is to count each one in the selected area. The density of the population is then obtained by dividing the number counted by the area. This method is the most direct, but it may also be extremely time consuming and difficult for those doing the counting, if not impossible. Populations of wildlife particularly those in the marine environment can only be estimated. One method for estimating wildlife populations is called line transect sampling. A general overview of the current theory is provided by Burnham et al. (1980) and is briefly described here.

The basic concept of line transect theory is demonstrated by considering an area of size **A** with known boundaries containing an unknown number of animals, (**N**). In order to estimate the number of animals in the area using line transect theory at least one line of travel must be established through the area. The animals are randomly distributed in the area to be sampled and the line of length **L** running through the area is randomly located with respect to the distribution of the animals, Figure 3.1. It is this randomness of placement that justifies extrapolating results to an area larger than that of the sampled area.

An observer then traverses this line counting the number of animals detected and their perpendicular distance from the transect line. The significance of this method is that not all the animals are detected and the animals nearest the transect line are more likely to be detected.

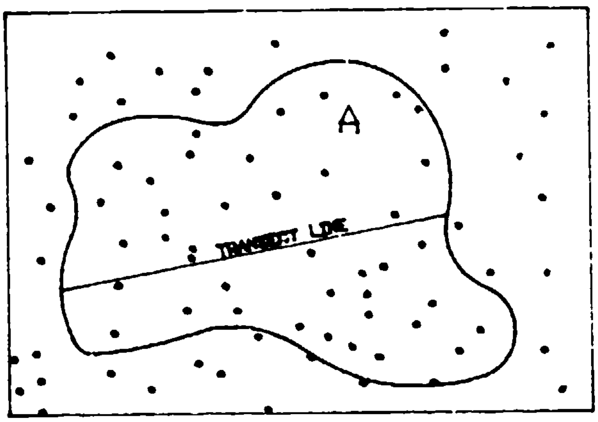



FIGURE 3.1 - Area **A** containing animals and transect line of length **L.**

There are four critical assumptions used in line transect theory. These are:

1) Animals or objects on the transect line are always detected;

2) Animals are fixed at the initial sighting position and do not move thereafter and no animal is counted twice;

3) Distances and angles are measured exactly with no rounding error;

4) Sightings are independent events.

The potential for a violation of any of these assumptions must be minimized.

The data collected from a line transect survey includes the number of animals detected in the area and their perpendicular distance, **x**, from the line transect, Figure 3.2. A model is then required that relates the collected data to the population density. The basic concept of the model is that the probability of an animal being detected by the observer decreases as its perpendicular distance from the transect line increases. This may be represented by a function or curve, **g(x)**, called a detection function.

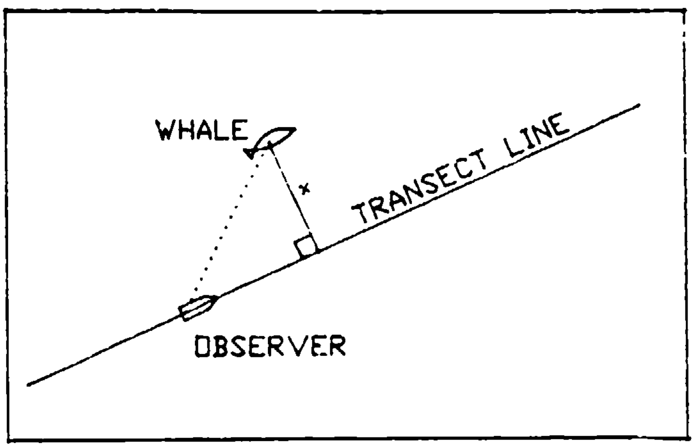



FIGURE 3.2 - The observer traveling along the transect line determines the angle and distance to the animal. From this information the perpendicular distance, **x**, between the animal and the transect line is obtained.

The detection function, **g(x)**, is defined as the probability of detecting an animal given its perpendicular distance from the transect line, **x**, or in terms of probability notation:

$$g(x) = P\ \{\ \text{object detected} \ / \ x\ \} \tag{3.1}$$

The first critical assumption of line transect theory is that if an object is on the transect line its probability of detection is perfect or equal to one, that is **g(x)** = 1. Then as **x** increases,

**g(x)** decreases. Examples of detection functions or curves are provided in Figure 3.3. It does not matter on which side of the transect line an animal is located, since it is assumed that data are analyzed without reference to either side of the line. Data from both sides of the line are pooled together.

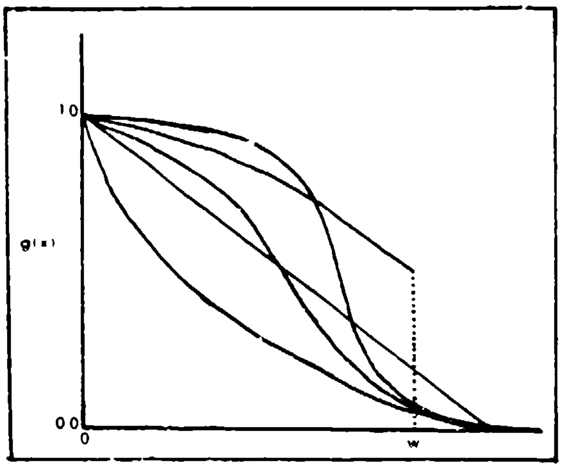



FIGURE 3.3 - Several possible shapes of the detection curve, **g(x)**. For one of the curves data is recorded only to distance **w**, this is called a truncation (Burnham et al., 1980).

The measured perpendicular distances, $\mathbf{x}_1 \ldots \mathbf{x}_n$, are directly related to the detection function. If an extremely large data sample were collected and the distance data was plotted on a histogram, the detection function is obtained by drawing a smooth curve through the bars of the graph as shown in Figure 3.4. In reality, however, sample sizes are not large enough to provide a smooth evenly distributed histogram, as shown in Figure 3.5.

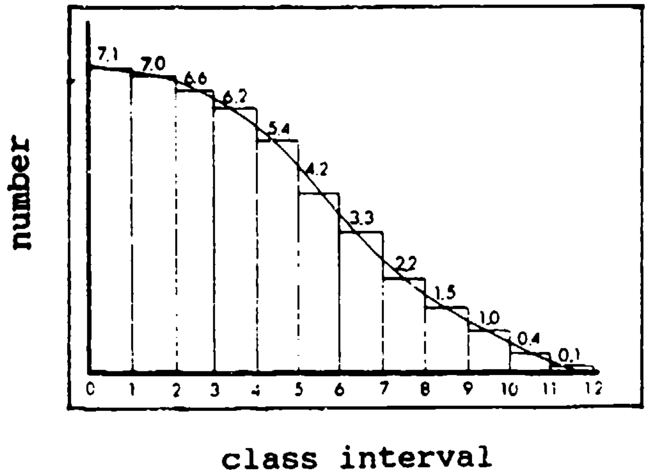



FIGURE 3.4 - Expected histogram of perpendicular distance data if **g(x)** has the shape shown for a sample size of n = 45. The vertical axis is the number of individuals (Burnham et al., 1980).

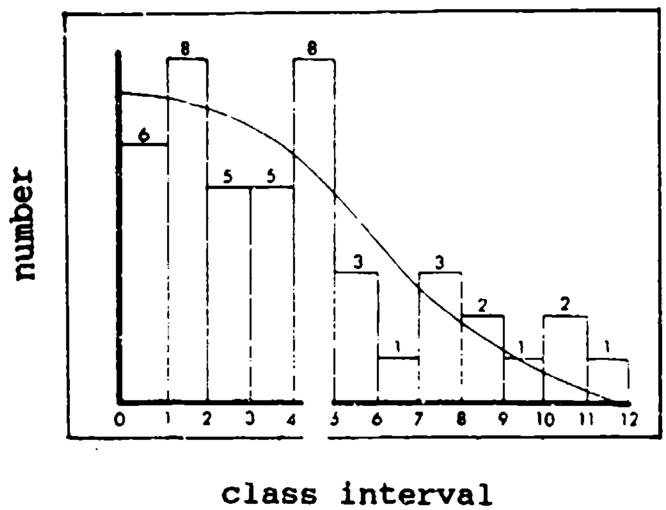



FIGURE 3.5 - Histogram from actual perpendicular distance data for one sample size of n = 45. The vertical axis is the number of individuals (Burnham et al., 1980).

The estimation of the density of animals in an area is given by

$$D_{\bullet} = n / 2La \tag{3.2}$$

where $D_{\bullet}$ is the estimated density per unit area, **n** is the number of

animals and $L$ is the length of the transect line. The unknown variable, $a$, may be expressed as

$$a = \int_0^w g(x)\,dx \tag{3.3}$$

where $w$ is the maximum perpendicular distance in which observations are made. The value of $w$ may be infinite or may be set at a specific distance.

The probability of detecting an animal is

$$P = 1/w \int_0^w g(x)\,dx \tag{3.4}$$

when $w$ is of finite value and the area of observation is

$$A = 2Lw \tag{3.5}$$

and the average probability of detecting an animal in the area is

$$P = a/w \tag{3.6}$$

If $N$ is the total number of objects in the sampled area then the

expected number of detected animals, $\mathbf{E(n)}$, is

$$\mathbf{E(n) = NP} \tag{3.7}$$

An estimation of $\mathbf{N}$ is then

$$\mathbf{N = n/P = nw/a} \tag{3.8}$$

The density in turn may be estimated by

$$\mathbf{D = N \;/\; A = N \;/\; 2Lw = nw \;/\; 2Lwa = n \;/\; 2La} \tag{3.9}$$

As shown in Equation 3.9, $\mathbf{D}$ is not a function of $\mathbf{w}$, so it does not matter if $\mathbf{w}$ is infinite or finite. If $\mathbf{a}$ is known, or rather if $\mathbf{g(x)}$ is known, then the density of animals may be calculated.

Strip transects are a special type of line transect. If $\mathbf{w}$ is determined to be finite, and it is assumed that all animals are detected on both sides of the transect line, then the line transect is called a strip transect. Figure 3.6 demonstrates line transect sampling when $\mathbf{w}$ is the finite boundary of the area to be searched. In a strip transect every animal in the area $\mathbf{2Lw}$ is assumed to be counted, implying a detection function of $\mathbf{g(x)} = 1$. The estimated density from a strip transect is then

$$\mathbf{D = n \;/\; 2Lw} \tag{3.10}$$

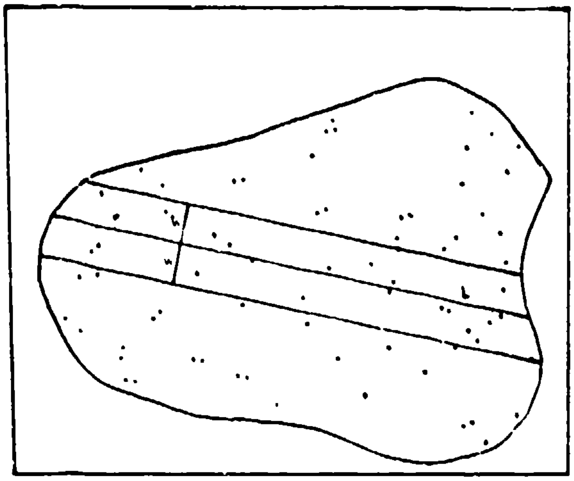


FIGURE 3.6 - Representation of line transect sampling when a finite boundary, w, is placed on the area to be searched. Only the points detected within the finite boundaries are recorded (Burnham et al., 1980).

Line transect theory is applicable for counting many species in many different environments. However, the theory described is for a two dimensional environment, either on land or at the surface of the ocean and depends on visual observation of detection of animals. Currently whale populations are being estimated using line transect theory, but this is for visual sightings at the surface (Cooke, 1985).

The actual method used to detect the whale and determine its distance from the transect line is not part of line transect theory. Therefore detection and location acoustically can be considered no different than that done visually. However, since line transect theory is designed only for two dimensions, it can not be used in its current form for animals detected acoustically. For animals such as sperm whales and other cetaceans a modification of the theory is needed to include the third dimension.

# Chapter 4

# Computer Simulation for Acoustical Estimation of Sperm Whale Abundance

## 4.1 Program Background and Structure

Line transect theory was originally intended for a two-dimensional visual environment. Prior to using the theory in a three-dimensional acoustic environment an analysis of the theory in this new environment is required. A computer program designed to simulate real conditions is one method. The computer program presented here, entitled "Line Transect Simulation - One" is constructed to represent real animals moving in a real three-dimensional environment. The animals simulated are sperm whales and the three-dimensional environment is the ocean. The computer program language is QBASIC.

The development of the computer simulation is based upon a few basic hypotheses that follow. The first hypothesis is that a given number of whales are moving about randomly in a volume. A line is traversed across the surface of the volume, simulating a ship towing a linear hydrophone array. As the line is traversed or as the ship moves across the surface the program determines if there are any whales within the specified range of the hydrophone array. If a whale is within range, a detection function is used to

determine the probability of the whale being detected while it is in the range of the hydrophones. If the whale is detected this is recorded and the ship continues to travel along the transect line.

The program can simulate in a two-dimensional or three-dimensional environment so that a comparison may be made between the two. The simulation is different from the real world environment because the detection function is known before any data is collected. In the real world the detection function must be assumed or calculated from data.

The program is segmented into subprograms each having a special task within the main program. Arrays of records are used to store data concerning each whale and its status during each program run. Graphics are used to display the program while it is running to give the user a visual interpretation of the whale and ship movement, but the program may also be run without graphics. The relative time unit is a minute, that is, the boat and the whales move to a new position every minute.

## 4.2 Whale and Ship Movement

The most important aspect of the program is representing the whales as realistically as possible including their diving patterns and acoustic behavior. In accordance with line transect theory, the whales must be randomly located with respect to the transect line.

The program first randomly locates a user specified number of whales in a volume of space 500 km across by 500 km wide and at a depth between 0 and 2 kilometers. Each whale is then located by three coordinates, identified by the variables, x, y, and z. These coordinates are relative to a coordinate axes at one corner of the area. The limiting depth of the area is 3 kilometers, so that at the "ocean" surface the z coordinate is equal to 3 kilometers and at the bottom is zero, Figure 4.1.

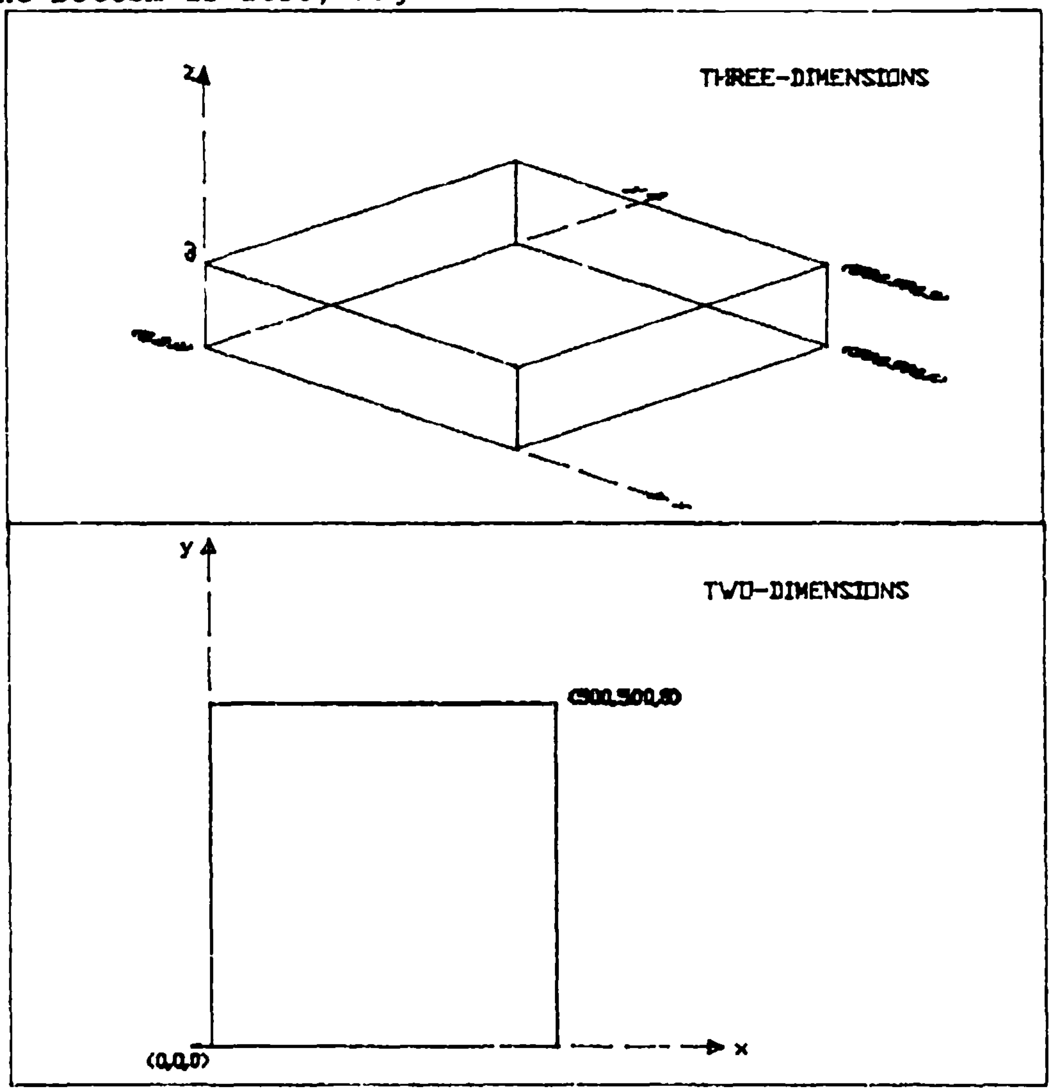



FIGURE 4.1 - Representation of the computer simulated ocean in two-dimensions and three-dimensions.

The depth of the simulated ocean is set at 3 kilometers to represent the Gulf of Mexico, since the Gulf is about 3 kilometers deep in most places away from the continental shelf. Although the whales are not located deeper than 2 kilometers, the program could be altered in the future to allow the whales to go deeper than 2 kilometers.

Besides having three locating coordinates, each whale is assigned additional information such as its diving depth, time underwater without coming to the surface, whether the whale is diving or surfacing, whether the whale has bee detected, and a velocity in the x and y coordinate directions. Once the whales are located and the program begins to run, the whales move about randomly within the area. The x and y velocity of each whale varies randomly between zero and a maximum value of about 13 knots or 24 kph. This is done by generating a random number between zero and 24. The diving and surfacing speed in the z direction is specified within the program and is identical for all whales. The diving speed is set to 3.5 knots or 6.4 kph and the surfacing speed is set at 5.2 knots or 9.6 kph.

The diving pattern of each whale is based on its diving depth and a sixty minute diving cycle. Each whale has its own 60 minute cycle. The first 15 minutes of the cycle are spent at the surface, the other 45 minutes are spent diving, swimming at the diving depth, and surfacing, Figure 4.2. The program is set so that if

the diving depth of the whale is too deep for the whale to reach and return to the surface in 45 minutes, the whale stops diving at a depth such that it can return to the surface before the end of the 60 minute cycle.

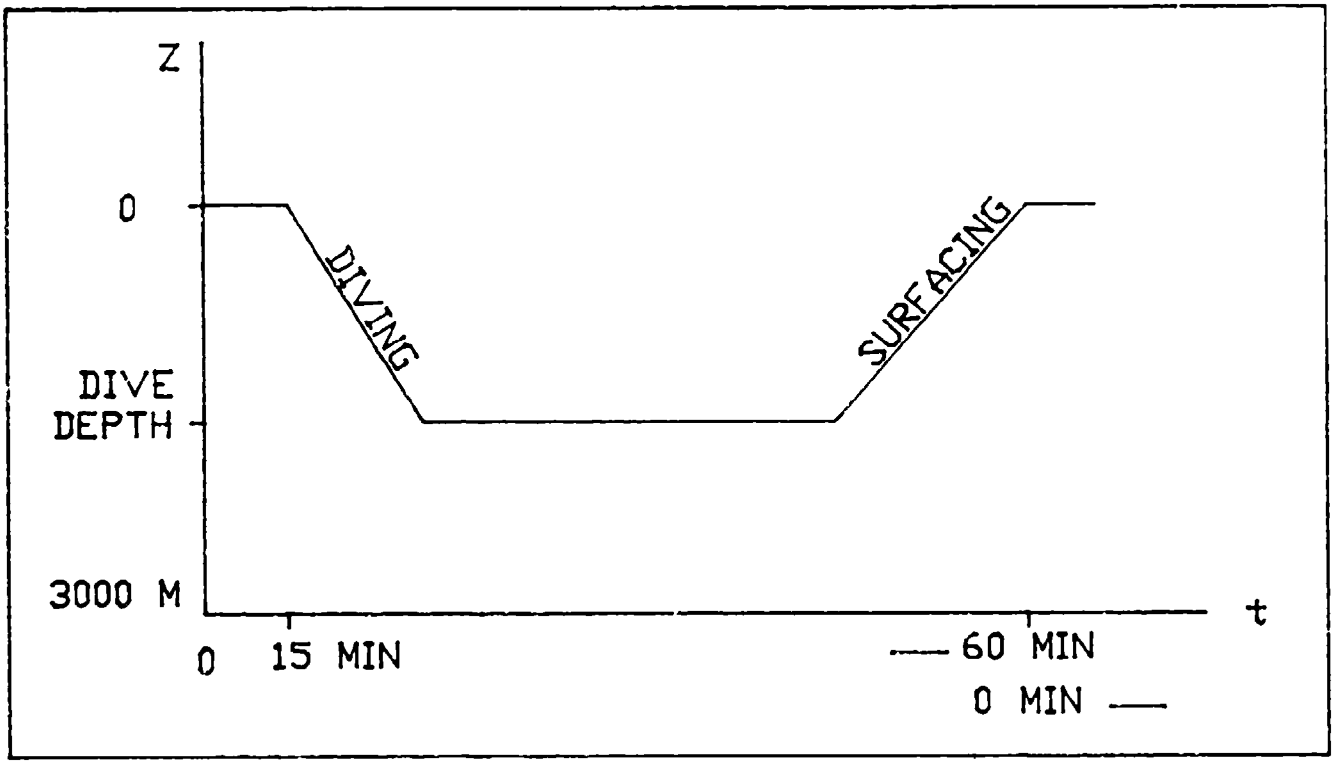



FIGURE 4.2 - Simulated sperm whale diving pattern.

For instance, at the start of the program one whale may be located at a depth of 500 meters and it is diving. Since the whale is not at the surface, it is in the middle of its cycle. As the program begins to run, the whale continues to dive deeper until the particular dive depth of that whale is reached. Once the whale has reached its diving depth, it stays at that depth, moving at its random velocity, until it reaches a time at which it must begin to surface in order to get to the surface before the end of 60

minutes. When the whale reaches the surface its cycle time is reset to zero and it spends the next 15 minutes at the surface.

If the diving depth of the whale is very deep, say 2000 meters, the whale may not be able to reach this depth and return to the surface at its current diving and surfacing speeds. In this case the whale is programmed to dive as deep as it can in the time allowed. Then, it stops diving and starts surfacing such that it will reach the surface at the end of its 60 minute cycle, Figure 4.3.

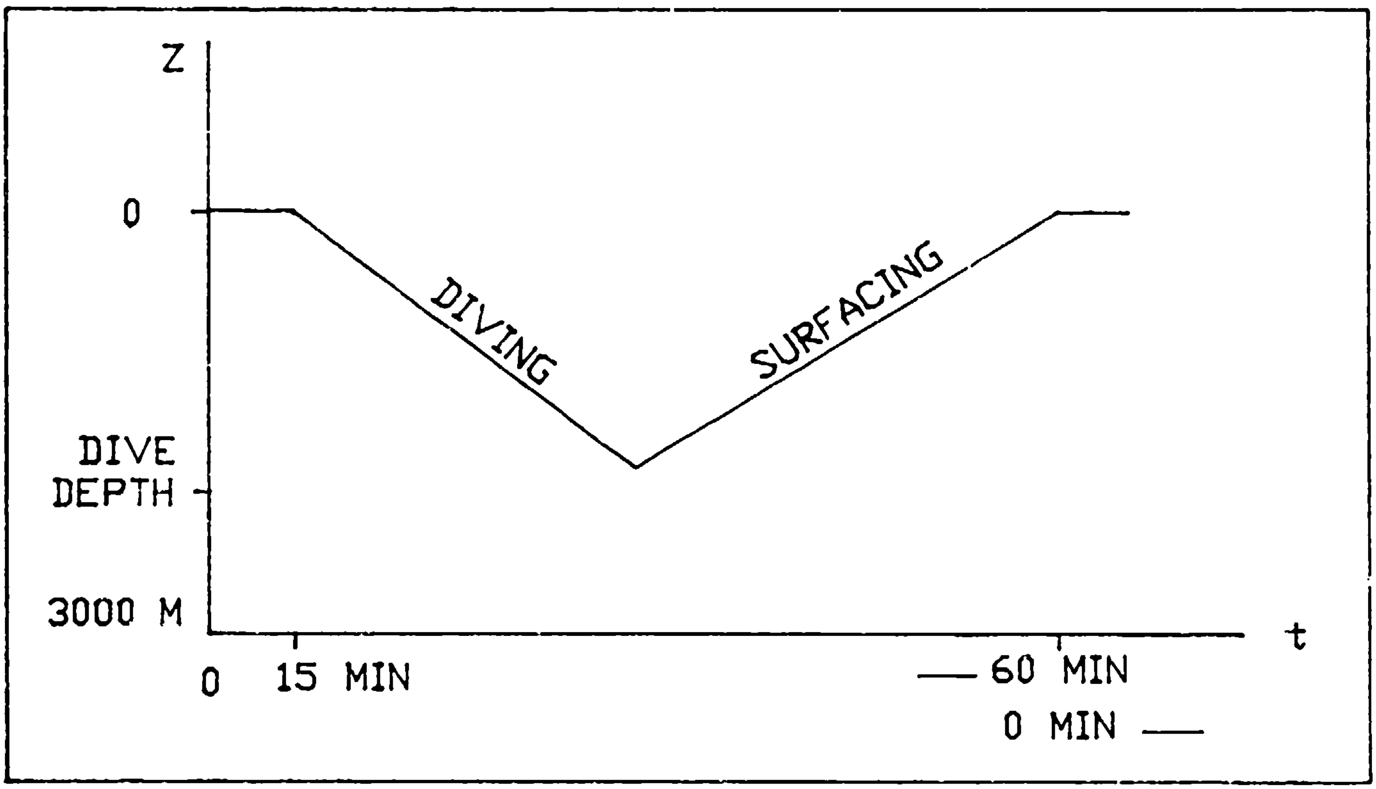



FIGURE 4.3 - Simulated sperm whale diving pattern when the shales diving depth can not be reached by the whale within the time limit in order to return to the surface in 60 minutes.

When the user specifies a certain number of whales, each is located randomly. This is done by generating a random number between 0 and

500 for the x-coordinate and y-coordinate and between 1000 and 3000 for the z-coordinate. At the start of the program each whale may be at the surface, at its own individual diving depth, or approaching its diving depth or surfacing. In other words, the whale may be located anywhere within the 60 minute cycle. Each whale will continue in its own particular diving cycle for as long as the program runs.

The transect line cuts diagonally across the simulated ocean from the origin of the three coordinate axes at x = 0, y = 0 and z = 3 kilometers to the opposite corner at x = 500 kilometers, y = 500 kilometers and z = 3 kilometers. The simulated ship towing the hydrophone array traverses the transect line at a speed specified by the user, Figure 4.4. Although the user inputs a speed in knots, this is converted to kilometers per minute for use within the program.

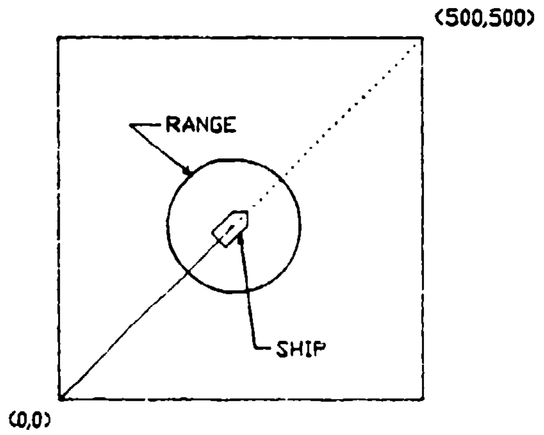



FIGURE 4.4 - Plan of two-dimensional view of the movement of the towing ship across the simulated ocean.

## 4.3 Detection of whales

The intent of the program is to tally the number of whales which are detected given a certain probability detection function, **g(x)**, discussed previously. In the case of underwater audio detection, sound sources may be located in any direction from the receiving hydrophone array. The program is an idealized version of the real world so the receiving equipment on the simulated ship is omnidirectional, able to receive signals equally from all directions and the water in the simulated ocean is homogeneous. Therefore, the acoustic range may be represented by a circle in two dimensions or a hemisphere in three dimensions. Any whale which is located within the circle or hemisphere has a chance of being detected.

The user determines if the program is run using the two-dimensional (2-D) case or the (3-D) three-dimensional case. In the two dimensional case, the depth location coordinate of the whale is excluded. All the whales are projected onto a flat plane and the area of detection is a circle with a radius equivalent to the user specified acoustic range. In the three-dimensional case the whales move about in a volume and have three locating coordinates, the two-dimensional coordinates, x and y, and an additional three-dimensional coordinate, z for depth. The three-dimensional volume of detection is a semisphere with a radius equivalent to the user specified range.

As the program runs, the ship traverses diagonally across the simulated ocean while the whales swim and dive in random directions. Moving along with the ship is the circle or hemisphere representing the acoustic detection range of the hydrophones towed by the ship. Whale detection does not begin until the ship has moved far enough so that any interference from the "sides" of the ocean are minimal. Similarly, whale detection ends before the ship enters the "corner" of the ocean and side effects interfere with the whale movements.

After every minute of program time, the ship stops and the program checks the location of every whale to determine if any are located within the acoustic detection range. If a whale is within the detection range a series of program decisions are made. First, the perpendicular distance between the whale and the transect line is calculated, as shown in Figure 4.5. This distance corresponds to a probability of detection from the detection function array. A random number between zero and one is generated for the whale. This random number is compared to the probability of detection. If the random number is less than the probability of detection then the whale is considered detected. If the random number is greater than the probability of detection then the whale is not detected. A whale can be detected only once while it is within the acoustic detection range. However, once it moves out of range it can be detected again if it later moves within the acoustic range a second time. If a whale is detected twice it can be easily seen in the

program output. In the field it is usually not known when a whale is detected twice along a transect unless that individual can be identified by some personal marking or characteristic. By allowing the previously detected whales to be detected again during a program run the user is able to see how often this is likely to occur in the field.

The probability function theoretically includes all the effects of the underwater environment such as spreading, attenuation, and background noise. The behavior of the whales, though, is part of the program, for example the diving pattern of the whales reflects the diving behavior of the sperm whale. Also, if a whale is within 5 meters of the surface it can not be acoustically detected because, according to studies of sperm whale behavior, sperm whales do not vocalize very near or at the surface.

The detection function is input into the program as a separate file. A mathematical formula is preferred to describe the detection function. However, the detection function tends to reflect the complicated, random events of nature and so it is usually an extremely complex function that is difficult to describe by a simple mathematical formula. Instead, the detection function is represented by a series of values or probabilities. Each probability corresponds to a perpendicular distance from the transect line. In this way, even very complex detection functions are approximated.

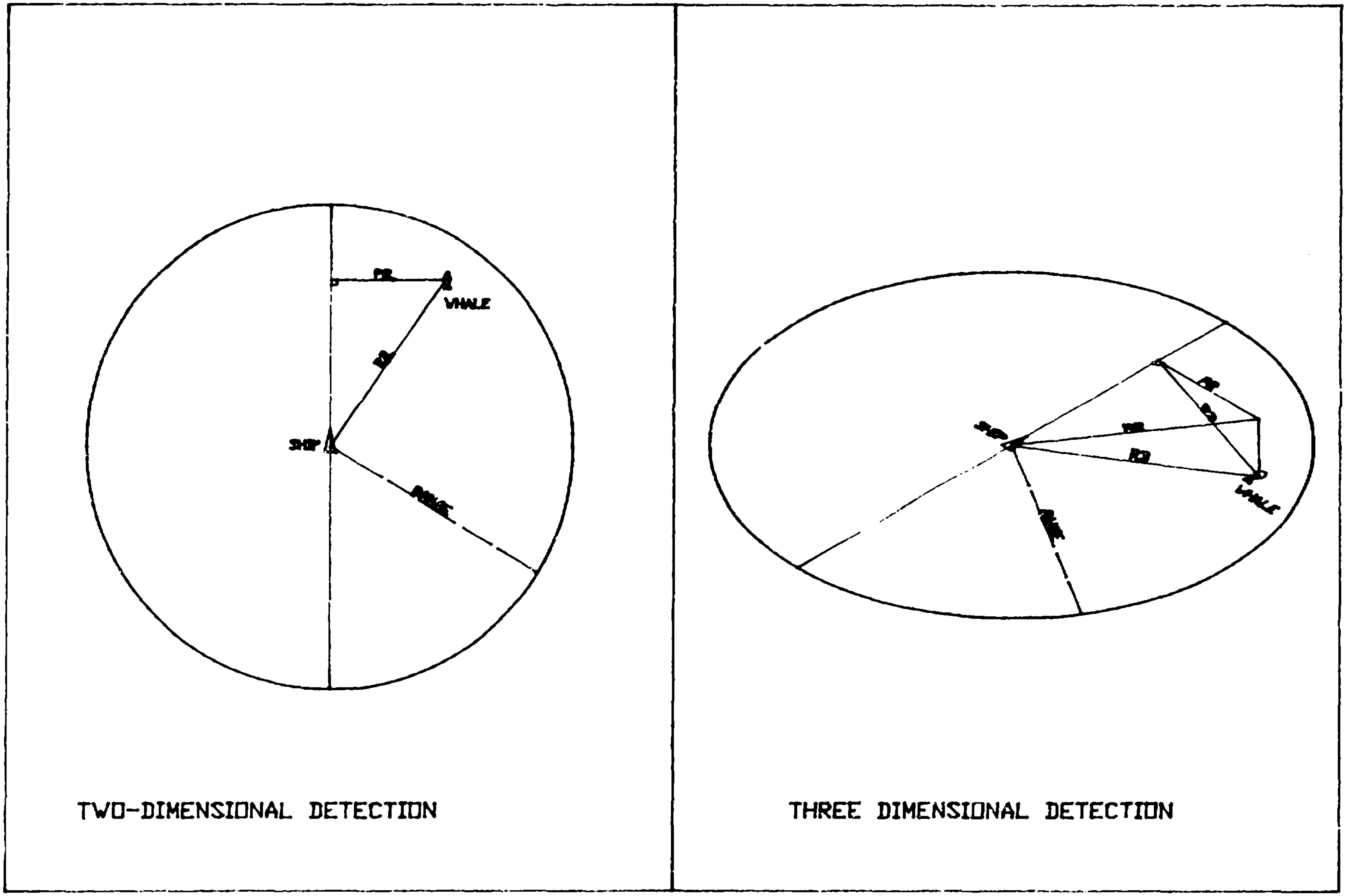



FIGURE 4.5 - Detection of a whale, in two-dimensions and in three-dimensions.

## 4.4 Input and Output

The program begins with an opening screen requesting the user to choose the graphic or nongraphic version of the program. Then there is a series of requests for input. The number of whales, the range of the acoustical equipment, and the speed of the towing ship are all needed. The user must decide if the program is to run in the two-dimensional mode or the three-dimensional mode.

There must be a file containing the values for the detection function, g(x), available for program input. The name of this file is requested and it must be written in ASCII code. Each probability must be located on a separate line. The first line starting with the probability of detection for an animal located on the transect line where x = 0 and the probability or g(x) = 1, and descending to the probability of detection for an animal at the edge of the acoustic range. There must be 101 probabilities in the file. Examples of detection functions are provided in Chapter 5, Figures 5.1 through 5.3. The detection function file may be located on drives A or B but this, with any path names, must be specified.

The program automatically sends output to the screen for viewing as the program runs. If the user has chosen the graphical version then a two-dimensional representation of the simulated ocean, whales and towing ship are included. User input is displayed along

file contains the information shown on the computer screen. Another has two pieces of information, the total number of whales and the number of whales detected during each program simulation run. The last file contains just the perpendicular distances of the detected whales. This file is useful in compiling a histogram of the perpendicular distances for comparison against the original input detection function curve.

# Chapter 5

# Computer Simulation Results and a Comparison to Field Measurements

## 5.1 Detection Functions and Testing the program

The program was tested using three different detection functions. These are described as Detection Function A, B and C and are shown in Figure 5.1 through Figure 5.3 respectively. For each detection function the program is run in both the two-dimensional mode and the three-dimensional mode, Cases 1 - 8. In addition, the Detection Function A is used to test the effect of varying input data in the two-dimensional mode.

In every test case the density of whales is held constant at 250 whales in an area of 250,000 $km^2$ or 0.001 whales per $km^2$. The speed of the towing ship and the acoustic range of the hydrophone array are maintained at values actually used in the field. Except for Case 2 the speed of the towing ship is maintained at 7 knots. Except for Case 3 the acoustic range is maintained at 5 km.

The amount of data generated by the program is considerable and is not included in this chapter, but a summary of the results for each test case is provided. A sample of the complete simulation results that may be obtained by the program is provided in Appendix C.

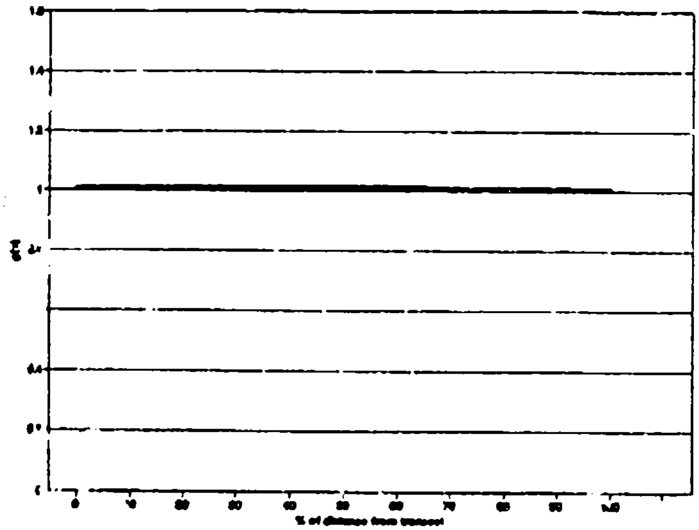


Figure 5.1 - Detection Function A where **g(x)** = 1 for all **x**.

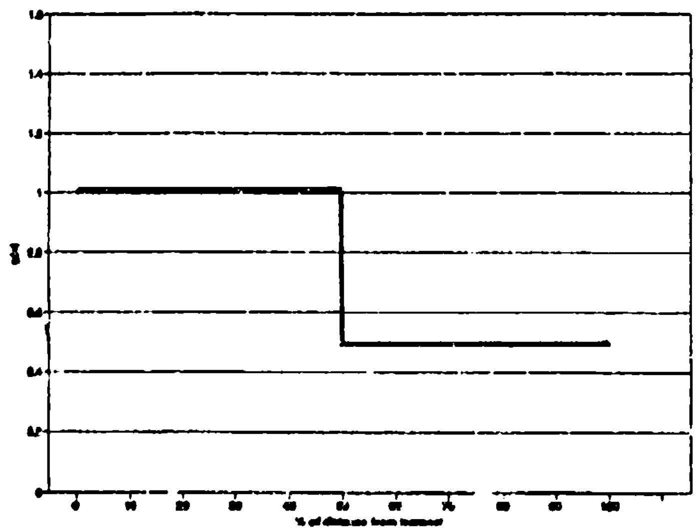


Figure 5.2 - Detection Function B where **g(x)** = 1 for all **x** <= 1/2 (range) and **g(x)** = 1/2 for all **x** > 1/2(range).

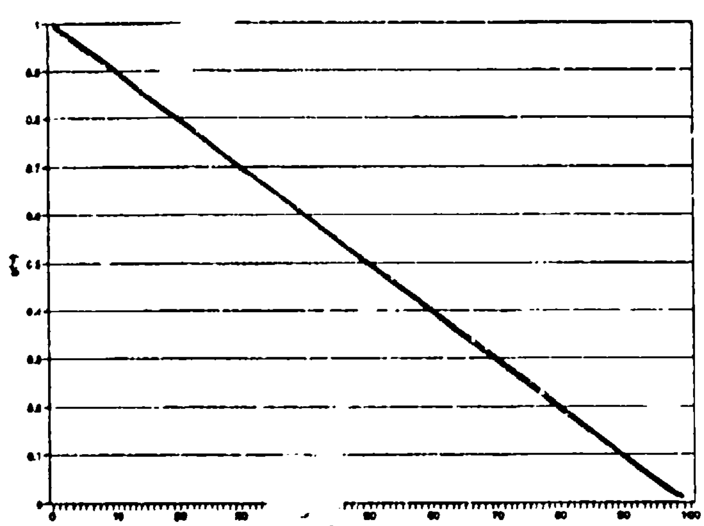


Figure 5.3 - Detection Function C, where **g(x)** = 1 - 0.2(**x**) for all **x**.

## 5.2 Verifying the Program

Case 1

The program was verified by using Detection Function A in which $\mathbf{g(x)} = 1$ for all $\mathbf{x}$ and the dimension mode is 2. In this case the probability of a whale being detected within the acoustic range is always 1. All whales, except those at or within 5 m of the surface, should be detected if they fall within the acoustic range of the ship towed hydrophone array. Since the density of whales is known, the result of the simulation using Detection Function A should indicate a density very close or slightly less then the known density. The input and output of Case 1 are shown in Table 5.1.

The length, **L**, of the transect line for one simulation program run is 500 km and the acoustic range, **w**, is 5 km. The known density, **D**, of whales is 0.001 whales/km$^2$. The number of whales that should be detected in any program run is calculated using

$$\mathbf{D = n / 2La} \qquad (5.1)$$

or

$$\mathbf{n = 2DLa} \qquad (5.2)$$

where **a** is determined as

$$\mathbf{a} = \int_0^w \mathbf{g(x)\ dx} = [\ \mathbf{x}\ ]_0^w = 5 \qquad (5.3)$$

The number of whales that should be detected in any program run in Case 1 is then

$$\mathbf{n} = \mathbf{2DLa} = 2 * (0.001) * 500 * 5 = 5 \text{ whales} \qquad (5.4)$$

As shown in Table 5.1 the average number of whales detected per run is 4.7 whales. Since the expected number of whales to be detected is 5 or slightly less than 5 the program does perform as expected in detecting whale population density.

Perpendicular distance data obtained by the simulation is compiled into intervals of one kilometer each as shown in Table 5.1. This compilation is used in the histogram in Figure 5.4. The expected number of whales to be detected per kilometer interval is obtained by dividing the total expected, 100, by the number of intervals, 5. This results in 20 whales per kilometer. The number of whales detected by the simulation in each kilometer interval does not quite meet the expected number of 20 whales per kilometer, as shown in Figure 5.4.

detection function: A
*****************************************
boat speed: 7 knots
range: 5 km
dimensions: 2

length of transect: 500 km
total number of whales: 250
number of runs: 20

expected number of whales to be detected: 100
expected number of whales to be detected per run: 5
expected number of whales to be detected per class interval: 20

total number detected in number of runs: 94
average per run: 4.7

| run number | detected | run number | detected |
|---|---|---|---|
| 1 | 4 | 11 | 2 |
| 2 | 5 | 12 | 7 |
| 3 | 4 | 13 | 4 |
| 4 | 9 | 14 | 7 |
| 5 | 4 | 15 | 5 |
| 6 | 8 | 16 | 3 |
| 7 | 1 | 17 | 7 |
| 8 | 6 | 18 | 4 |
| 9 | 3 | 19 | 5 |
| 10 | 7 | 20 | 2 |

| kilometer interval | number of individuals |
|---|---|
| 0.0 - 1.0 | 27 |
| 1.0 - 2.0 | 20 |
| 2.0 - 3.0 | 24 |
| 3.0 - 4.0 | 24 |
| 4.0 - 5.0 | 32 |

Table 5.1 - Input and Output of Case 1.

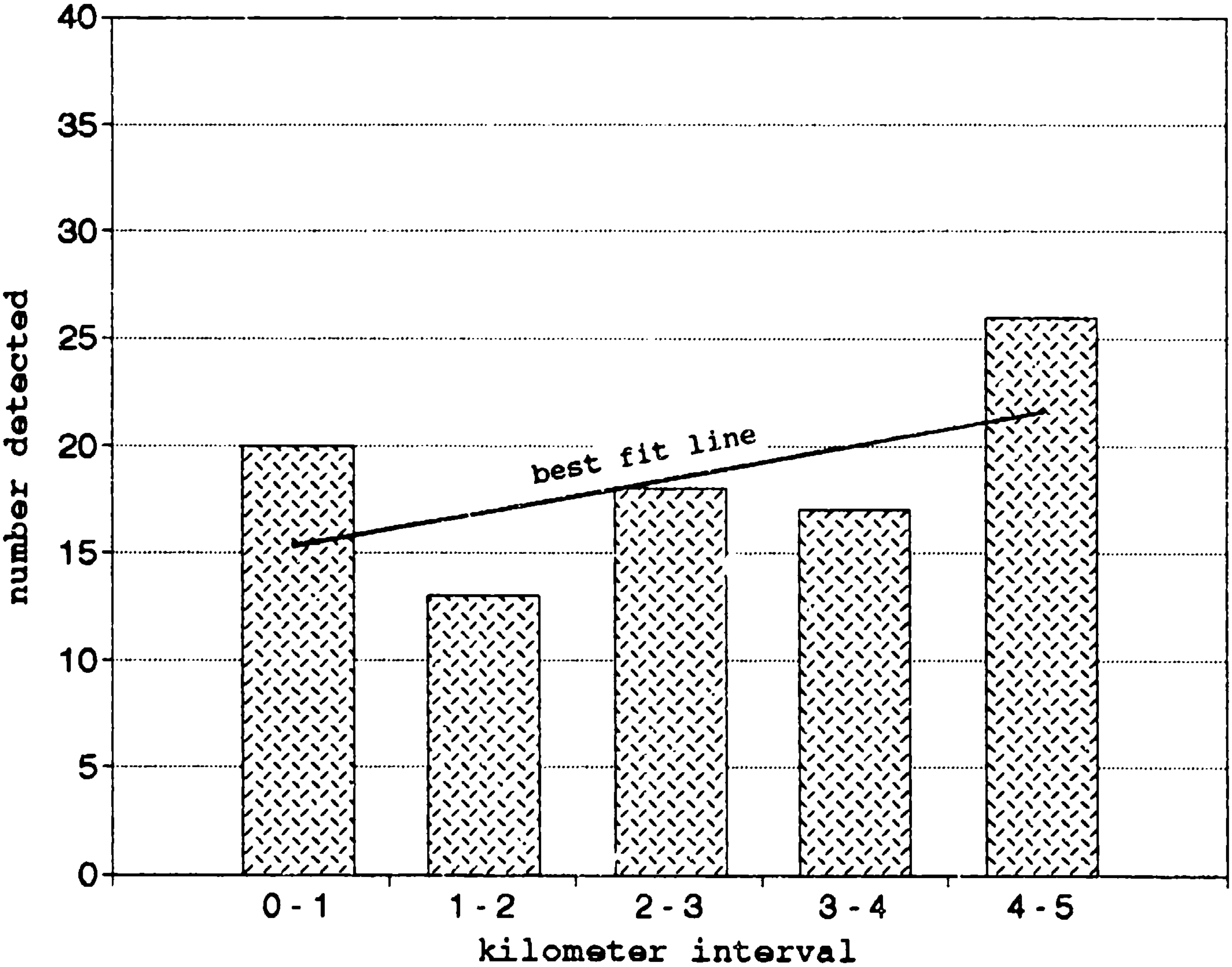



Figure 5.4 - Histogram for Case 1 using Detection Function A, n = 94.

## 5.3 Examples

Cases 2 and 3 are examples of the program behavior when Detection Function A is used as in Case 1, but the speed of the ship is increased - Case 2; or the acoustic range is increased - Case 3.

### Case 2

The input and output for Case 2 is shown in Table 5.2. The speed of the towing ship is 14 knots, double that of Case 1. This is not a realistic situation since the real towing ship rarely goes over speeds of 10 knots, particularly if it is towing a hydrophone array. The effect on the program is to lessen the number of increments along the transect line where a check for whales in the detectable range is performed. Therefore, less whales should be detected in Case 2 than there are in Case 1.

As shown in Table 5.2, the total number of whales detected in Case 2 is only 76 in 20 program runs compared 94 whales detected in Case 1. The average number of whales detected in Case 2 is 3.8 whales per program run. This is almost a whole whale less than the average detected in Case 1.

The histogram for Test Case 2 is shown in Figure 5.5. As in Test Case 1, the histogram results are erratic and do not reflect the shape of Detection Function A as shown in Figure 5.1.

detection function: A
******************************************
boat speed: 14 knots
range: 5 km
dimensions: 2

length of transect: 500 km
total number of whales: 250
number of runs: 20

expected number of whales to be detected: 100
expected number of whales to be detected per run: 5
expected number of whales to be detected per class interval: 10

total number detected in number of runs: 76
average per run: 3.8

| run number | detected | run number | detected |
|---|---|---|---|
| 1 | 1 | 11 | 5 |
| 2 | 1 | 12 | 2 |
| 3 | 6 | 13 | 7 |
| 4 | 3 | 14 | 2 |
| 5 | 3 | 15 | 4 |
| 6 | 4 | 16 | 5 |
| 7 | 3 | 17 | 5 |
| 8 | 5 | 18 | 2 |
| 9 | 4 | 19 | 4 |
| 10 | 3 | 20 | 7 |

| kilometer interval | number of individuals |
|---|---|
| 0.0 - 1.0 | 14 |
| 1.0 - 2.0 | 11 |
| 2.0 - 3.0 | 11 |
| 3.0 - 4.0 | 22 |
| 4.0 - 5.0 | 18 |

Table 5.2 - Input and Output of Case 2.

Case 3

The input and output for Case 3 are shown in Table 5.3. The acoustic range of the hydrophone array is increased to 10 km. This is not a realistic value since the present maximum range of the acoustic equipment in the field is only 5 km. By increasing the acoustic range the area that is checked for detectable whales increases, so more whales should be detected in Case 3 than there are detected in Case 1.

The length, **L**, of the transect line is still 500 km but the acoustic range, **w** is now 10 km. The known density, **D**, of whales is 0.001 whales/km$^2$. The number of whales that should be detected in any program run is calculated using Equation 5.2 where **a** is determined as

$$\mathbf{a} = \int_0^w g(x)\, dx = [\, x \,]_0^{10} = 10 \qquad (5.5)$$

The number of whales detected in any run in Case 3 is then

$$\mathbf{n} = \mathbf{2DLa} = 2 * (0.001) * 500 * 10 = 10 \text{ whales} \qquad (5.6)$$

As shown in Table 5.3 the number of whales per run detected by the simulation program is 12.75, exceeding the expected number of whales per program run by 2.75 whales.

detection function: A

******************************************

boat speed: 7 knots
range: 10 km
dimensions: 2

length of transect: 500 km
total number of whales: 250
number of runs: 20

expected number of whales to be detected: 200
expected number of whales to be detected per run: 10
expected number of whales to be detected per class interval: 20

total number detected in number of runs: 255
average per run: 12.75

| run number | detected | run number | detected |
|---|---|---|---|
| 1 | 11 | 11 | 14 |
| 2 | 12 | 12 | 17 |
| 3 | 18 | 13 | 13 |
| 4 | 15 | 14 | 19 |
| 5 | 12 | 15 | 16 |
| 6 | 11 | 16 | 15 |
| 7 | 17 | 17 | 11 |
| 8 | 10 | 18 | 3 |
| 9 | 8 | 19 | 11 |
| 10 | 10 | 20 | 15 |

| kilometer interval | number of individuals | kilometer interval | number of individuals |
|---|---|---|---|
| 0 - 1 | 26 | 5 - 6 | 22 |
| 1 - 2 | 18 | 6 - 7 | 24 |
| 2 - 3 | 18 | 7 - 8 | 19 |
| 3 - 4 | 32 | 8 - 9 | 31 |
| 4 - 5 | 30 | 9 - 10 | 35 |

Table 5.3 - Input and Output of Case 3.

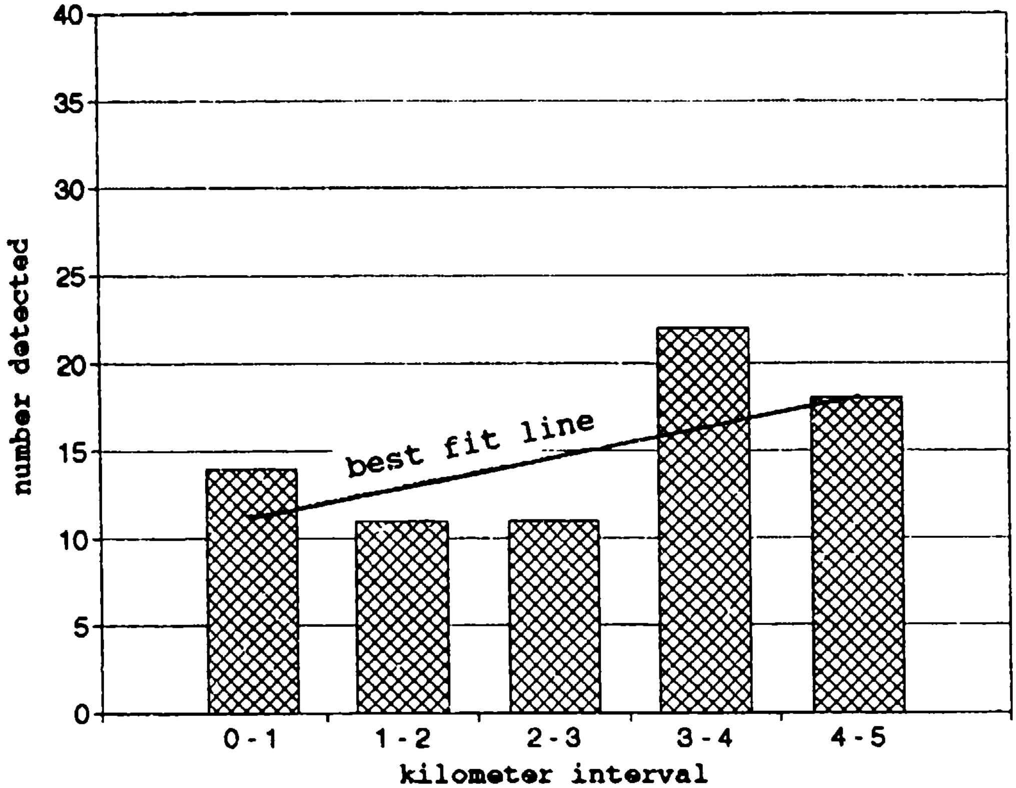



Figure 5.5 - Histogram for Case 2, using Detection Function A.

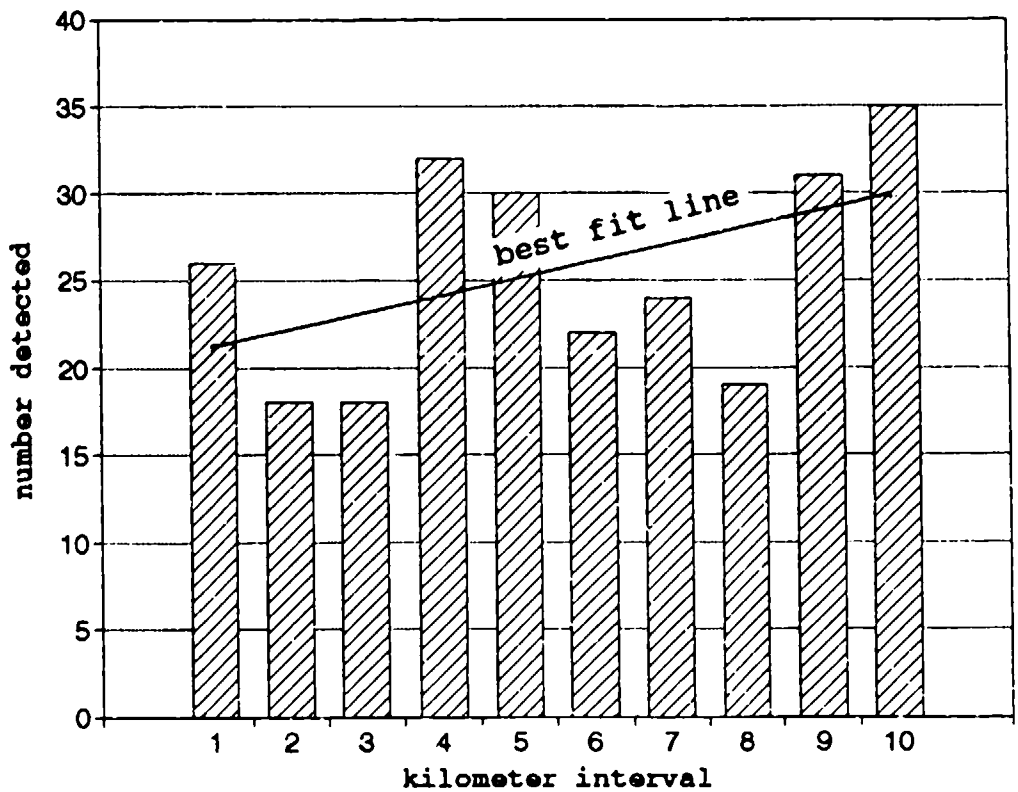



Figure 5.6 - Histogram for Case 3 using Detection Function A.

Case 4

In this test the input data remained the same as that in Case 1 but a different detection function was used, Detection Function B. For this detection function there is a probability of detection of 1 for all whales within 2.5 km of the hydrophone array. For whales located between 2.5 and 5.0 km the probability of detection is only 0.5 or half. The number of whales detected using this detection function should be less than using Detection Function A where $\mathbf{g(x)} = 1$ for all $\mathbf{x}$. There should also be an apparent drop in the number of whales detected beyond 2.5 km on the histogram.

Table 5.4 contains the input and output for Case 4 and a histogram of the perpendicular distance output data is shown in Figure 5.7. In order to see if there is a drop in the number of whales detected beyond 2.5 km, the histogram is divided into 4 intervals of 1.25 km each.

Using the same method as in Case 1, the value of $\mathbf{a}$ is determined to be

$$\mathbf{a} = \int_0^w g(x)\,dx = 2.5 + 1.25 = 3.75 \qquad (5.7)$$

and the expected number of whales to be detected per run is then

$$n = 2DLa = 2 * (0.001) * 500 * 3.75 = 3.75 \text{ whales} \quad (5.8)$$

Since there were 20 program runs completed for Case 4, the total number of detected whales is 75.

The number of whales detected by the simulation program in Case 4 is greater than that expected by almost one whale per run or 20% more of the expected value. In addition, the histogram, Figure 5.7, does not exhibit a significant drop in the number of whales detected beyond 2.50 km. However, there are less whales detected beyond the 2.50 km range in Case 4 when compar d to the results of Case 1.

detection function: B

**************************************************

boat speed: 7 knots
range: 5 km
dimensions: 2
length of transect: 500 km
number of whales: 250
density of whales: 0.001 whales/km2

number of runs: 20

expected number of whales to be detected: 75
expected number of whales to be detected per run: 3.75

total number detected in number of runs: 100
average per run: 5.0

| run number | detected | run number | detected |
|---|---|---|---|
| 1 | 3 | 11 | 3 |
| 2 | 3 | 12 | 5 |
| 3 | 3 | 13 | 6 |
| 4 | 4 | 14 | 2 |
| 5 | 5 | 15 | 3 |
| 6 | 6 | 16 | 5 |
| 7 | 4 | 17 | 5 |
| 8 | 7 | 18 | 9 |
| 9 | 4 | 19 | 12 |
| 10 | 8 | 20 | 2 |

| kilometer interval | number of individuals |
|---|---|
| 0.0 - 1.25 | 25 |
| 1.25 - 2.50 | 26 |
| 2.50 - 3.75 | 24 |
| 3.75 - 5.00 | 25 |

Table 5.4 - Input and Output of Case 4.

Case 5

In this test Detection Function C, Figure 5.3, is used. This function steadily decreases with distance from the point of $x = 0$. In this situation the expected number of whales detected should be much less than that detected in Case 1 because there is less probability for whales to be detected. For instance, the probability of a whale being detected in the interval between 4.0 and 5.0 kilometers is less than 0.10. Table 5.5 contains the results of 20 program runs and Figure 5.8 displays a histogram of the output data.

Using the same method as in Case 1, the value for **a** is determined to as

$$\mathbf{a} = \int_0^{\infty} g(x)\, dx = 2.5 \qquad (5.9)$$

The expected number of detected whales in any program run is then

$$\mathbf{n} = \mathbf{2DLa} = 2 * (0.001) * 500 * 2.5 = 2.5 \text{ whales} \qquad (5.10)$$

The results show that the number of whales detected by the program is more than double the expected value. However, the histogram for Case 5 when compared to the histogram for Case 1 does reflect a decrease in whales detected as the distance between the whale and the transect line increases.

detection function: C
*****************************************

boat speed: 7 knots
range: 5 km
dimensions: 2

length of transect: 500 km
total number of whales: 250
number of runs: 20

expected number of whales to be detected: 50
expected number of whales to be detected per run: 2.5

total number detected in number of runs: 98
average per run: 4.9

| run number | detected | run number | detected |
|---|---|---|---|
| 1 | 6 | 11 | 2 |
| 2 | 7 | 12 | 6 |
| 3 | 3 | 13 | 6 |
| 4 | 3 | 14 | 2 |
| 5 | 11 | 15 | 3 |
| 6 | 6 | 16 | 5 |
| 7 | 6 | 17 | 11 |
| 8 | 2 | 18 | 4 |
| 9 | 3 | 19 | 5 |
| 10 | 5 | 20 | 5 |

| kilometer interval | number of individuals |
|---|---|
| 0.0 - 1.0 | 35 |
| 1.0 - 2.0 | 16 |
| 2.0 - 3.0 | 20 |
| 3.0 - 4.0 | 18 |
| 4.0 - 5.0 | 9 |

Table 5.5 - Results of Case 5.

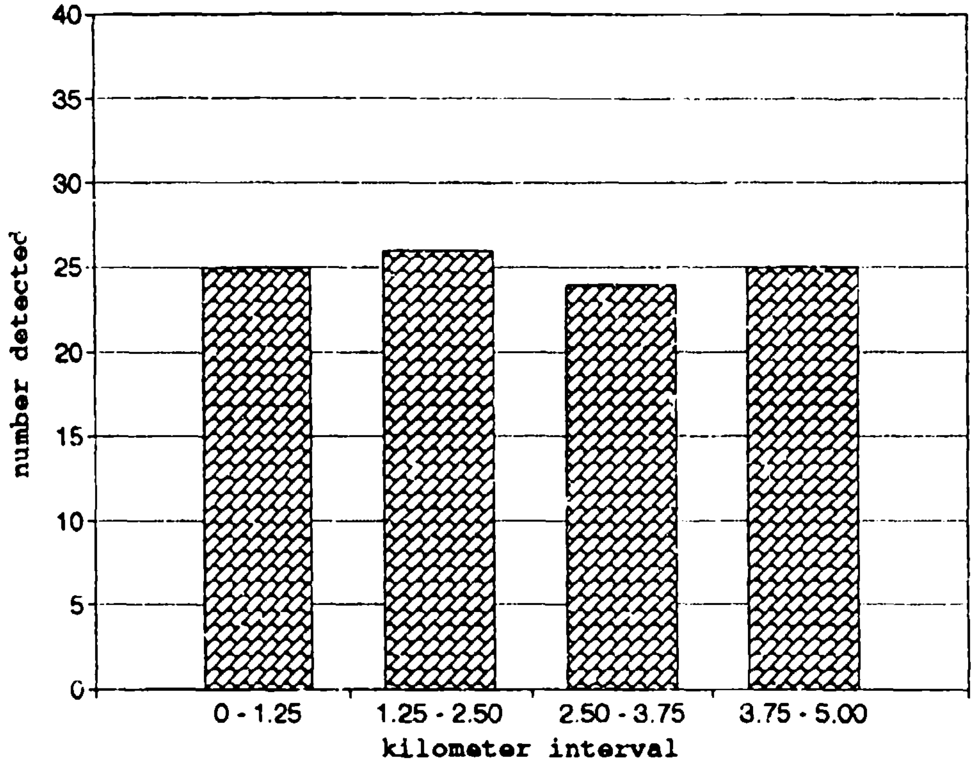



Figure 5.7 - Histogram for Case 4 using Detection Function B, n = 100.

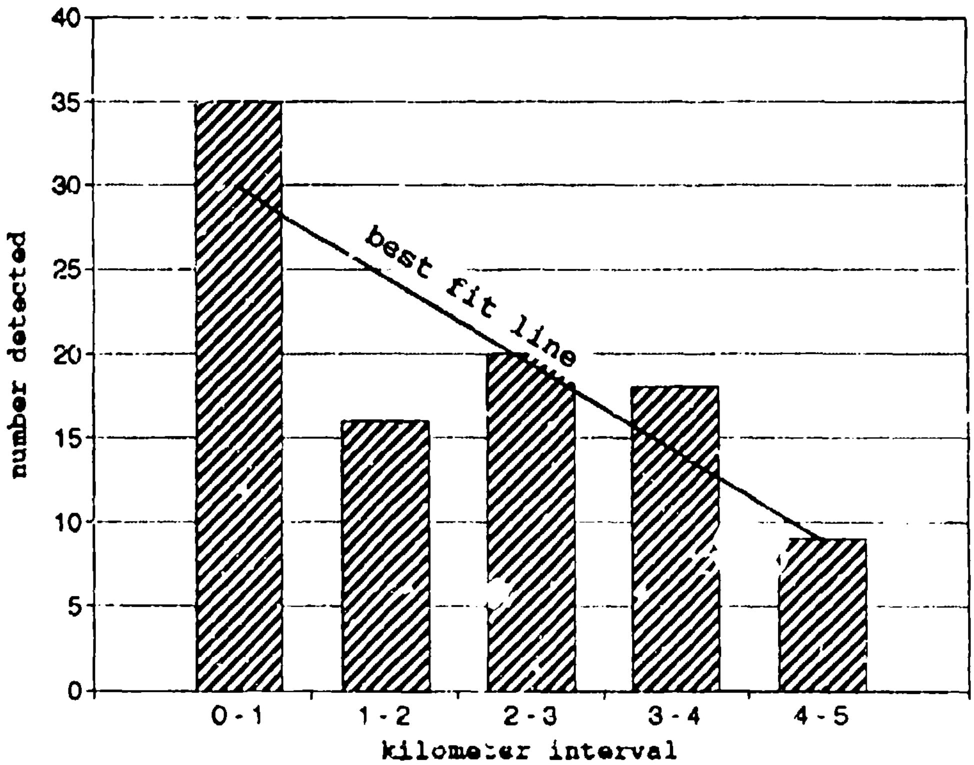



Figure 5.8 - Histogram of the results of Case 5, using Detection Function C, n = 98.

Case 6

In cases 6 through 8 the three-dimensional mode is used. Case 6 is the same as Case 1 except the dimension is changed to 3. The results of this case are shown in Table 5.6 and the resulting histogram is displayed in Figure 5.9. Since line transect theory is not developed for three dimensions a value for the expected number of whales is not calculated for Case 6. The case does show the difference between running the program with the same input in two different dimensions, two and three.

If the histogram from the two-dimensional simulation, Case 1, is compared to the histogram from the three dimensional simulation, Case 6, Figure 5.9, the significant difference between the two is located in the interval of 0 to 1.0 km. The three-dimensional mode shows fewer whales are detected near the transect line as compared to the two-dimensional mode. Both histograms indicate that more whales are detected when they are in the region farthest from the transect line between 4 and 5 kilometers.

In the three-dimensional case an increasingly larger volume is checked for whales as the distance from the hydrophone array increases. For instance, the volume checked for whales between 0 and 1.0 km is 12.5 $km^3$, but the volume checked for whales between 4.0 and 5.0 km is 766.5 $km^3$. The bigger volume means more whales may exist in the space between 4.0 and 5.0 km than in that between 0 and 1.0 km even though the interval between both is still 1 km.

detection function: A
**********************************
boat speed: 7 knots
range: 5 km
dimensions: 3
length of transect: 500 km

total number of whales: 250
density of whales: 0.00033 whales/$km^3$

number of runs: 20

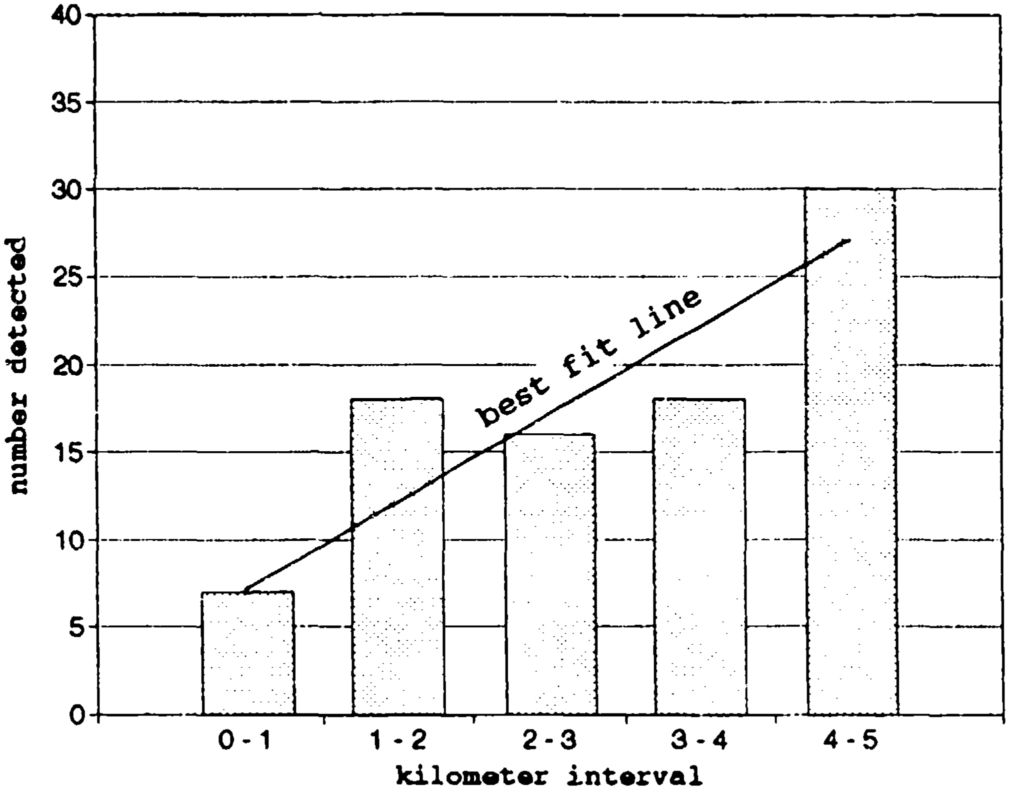
total number detected in number of runs: 89
average per run: 4.45

| run number | detected | run number | detected |
|---|---|---|---|
| 1 | 3 | 11 | 1 |
| 2 | 7 | 12 | 6 |
| 3 | 1 | 13 | 8 |
| 4 | 7 | 14 | 5 |
| 5 | 6 | 15 | 6 |
| 6 | 7 | 16 | 4 |
| 7 | 0 | 17 | 3 |
| 8 | 3 | 18 | 4 |
| 9 | 8 | 19 | 5 |
| 10 | 6 | 20 | 5 |

| kilometer interval | number of individuals |
|---|---|
| 0.0 - 1.0 | 7 |
| 1.0 - 2.0 | 18 |
| 2.0 - 3.0 | 16 |
| 3.0 - 4.0 | 18 |
| 4.0 - 5.0 | 30 |

---

Table 5.6 - Results of Test Case 6.



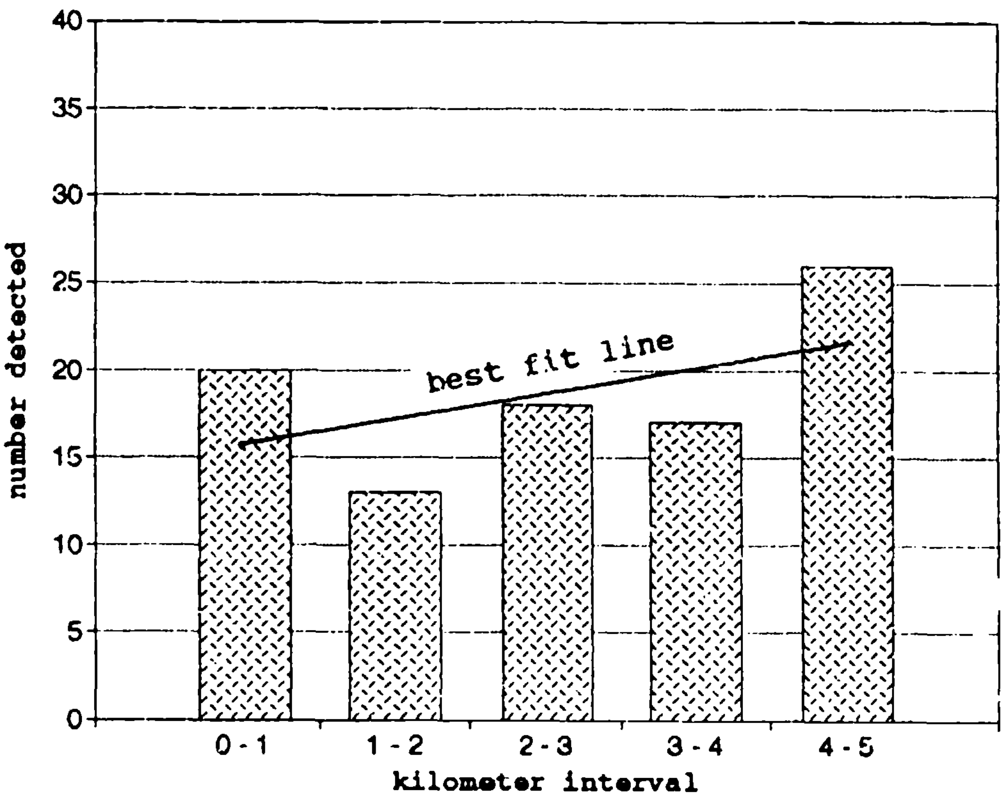




Figure 5.9 - Histogram for Case 6 (top), n = 89 compared to the histogram for Case 1 (bottom), n = 94.

Case 7

This test case is similar to Case 4 in that the input is the same, Detection Function B is used, but the three-dimensional mode is used instead of the two-dimensional mode. The input and output for Case 7 is shown in Table 5.7 and a histogram of the data is shown in Figure 5.10.

Comparing the histogram of Case 6, to the histogram of Case 7 a shift can be seen in the interval where the most whales are detected. In Case 6 more whales are detected near the edge of the acoustic range from 4 to 5 kilometers. In Case 7 the probability of detection past 2.5 km from the transect line is only half that within 2.5 km of the transect line. The most whales are detected in the acoustic range of 1.25 to 2.50 km. With the probability of detection still one near the transect line the number of whales detected in that interval is still as low as Case 6.

detection function: B
**********************************
boat speed: 7 knots
range: 5 km
dimensions: 3
length of transect: 500 km

total number of whales: 250
density of whales: 0.00033 whales/km$^3$

number of runs: 20

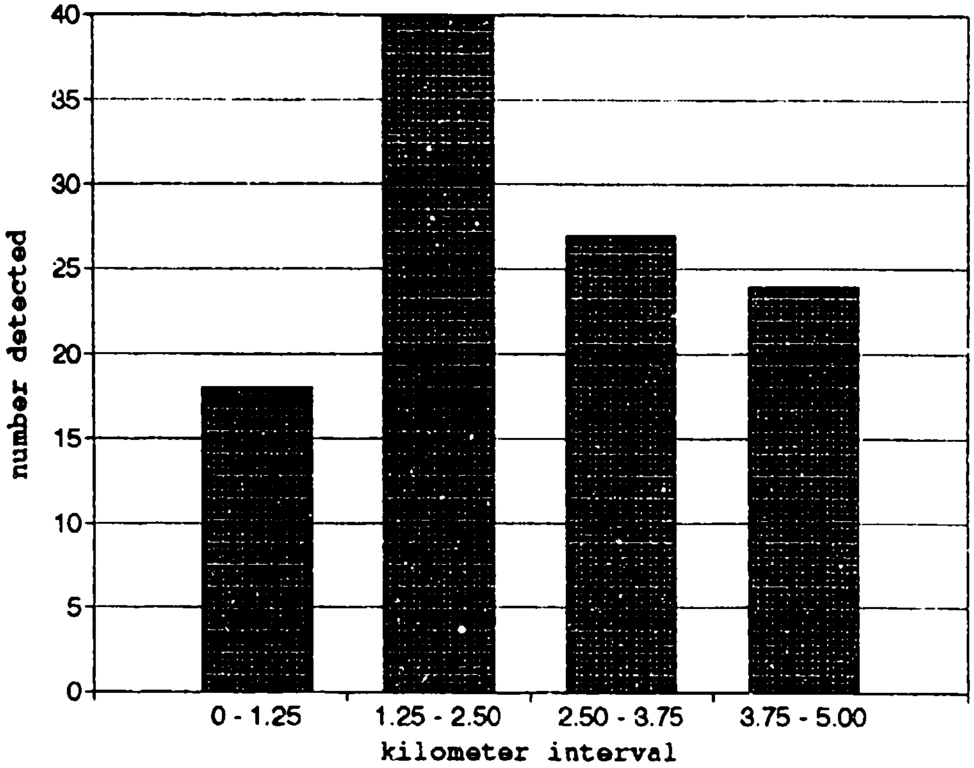
total number detected in number of runs: 87
average per run: 5.4

| run number | detected | run number | detected |
|---|---|---|---|
| 1 | 4 | 11 | 6 |
| 2 | 5 | 12 | 3 |
| 3 | 8 | 13 | 8 |
| 4 | 6 | 14 | 5 |
| 5 | 7 | 15 | 4 |
| 6 | 4 | 16 | 3 |
| 7 | 7 | 17 | |
| 8 | 9 | 18 | |
| 9 | 6 | 19 | |
| 10 | 2 | 20 | |

| class interval | number of individuals |
|---|---|
| 0.0 - 1.25 | 16 |
| 1.25 - 2.50 | 36 |
| 2.50 - 3.75 | 26 |
| 3.75 - 5.00 | 21 |

-----------------------------------------------------------------

Table 5.7 - Results of Case 7.



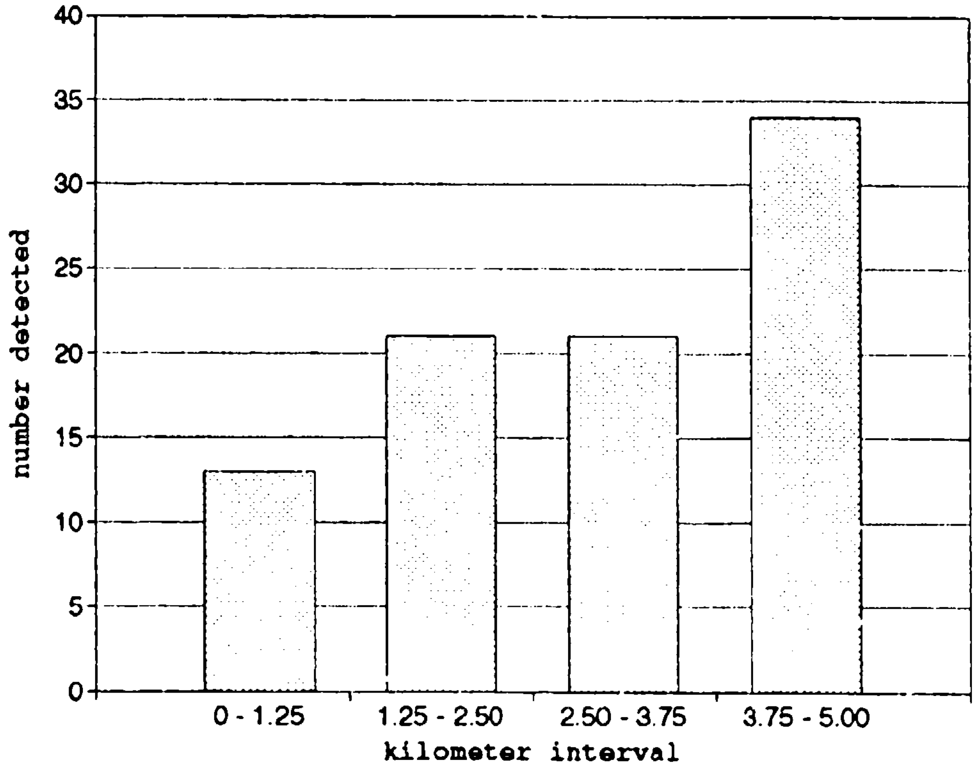




Figure 5.10 - Histogram from Case 7 results (top), using Detection Function B compared to histogram from Case 6 results (bottom).

Case 8

This case has the same input as Case 5 where the Detection Function C is used, except the simulation is in three-dimensional mode. The results of Case 5 are shown in Table 5.8 and the histogram of the data is provided in Figure 5.11. Even when using Detection Function C, in the three-dimensional mode there is still a tendency for less whales to be detected when they are very near the transect line even though the probability of detection is greatest at this distance.

Figure 11 contains two histograms. One histogram is a compilation of data after only 20 program runs and the other is after 34 program runs. By comparing the two histograms it is clear that the it is important to have a sufficient number of samples or in this case program runs in order to obtain a histogram reflective of the original detection function and to minimize the randomness of the data. The histogram produced after 20 program runs does not obviously reflect the detection function C, but after 34 program runs the data begin to mimic the original detection function.

detection function: C
**********************************
boat speed: 7 knots
range: 5 km
dimensions: 3
length of transect: 500 km

total number of whales: 250
density of whales: 0.00033 whales/$km^3$

number of runs: 20
total number detected in 20 runs: 84
average per run: 4.2

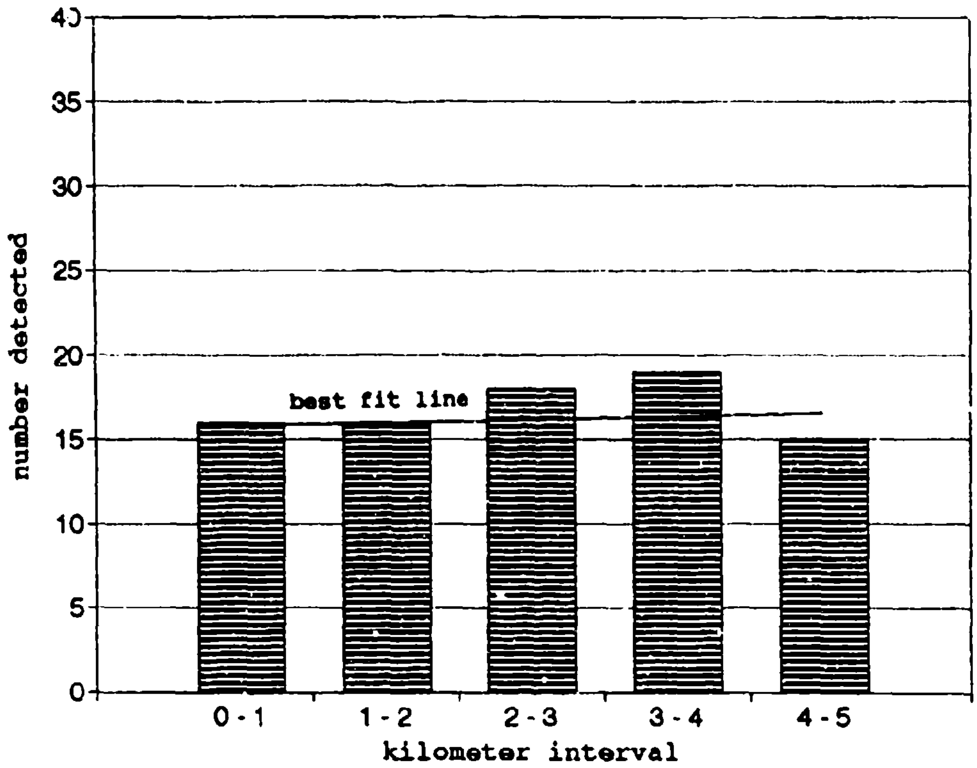
number of runs: 34
total number detected in 34 runs: 159
average per run: 4.7

| run number | detected | | |
|---|---|---|---|
| 1 | 5 | 18 | 6 |
| 2 | 2 | 19 | 7 |
| 3 | 5 | 20 | 1 |
| 4 | 6 | 21 | 3 |
| 5 | 6 | 22 | 11 |
| 6 | 2 | 23 | 7 |
| 7 | 6 | 24 | 5 |
| 8 | 1 | 25 | 4 |
| 9 | 7 | 26 | 8 |
| 10 | 0 | 27 | 3 |
| 11 | 5 | 28 | 12 |
| 12 | 3 | 29 | 5 |
| 13 | 5 | 30 | 9 |
| 14 | 7 | 31 | 3 |
| 15 | 5 | 32 | 0 |
| 16 | 0 | 33 | 4 |
| 17 | 7 | 34 | 6 |

| kilometer interval | number detected | |
|---|---|---|
| | after 20 runs | after 34 runs |
| 0.0 - 1.0 | 16 | 25 |
| 1.0 - 2.0 | 16 | 39 |
| 2.0 - 3.0 | 18 | 38 |
| 3.0 - 4.0 | 19 | 32 |
| 4.0 - 5.0 | 15 | 25 |

---

Table 5.8 - Results of Test Case 8.



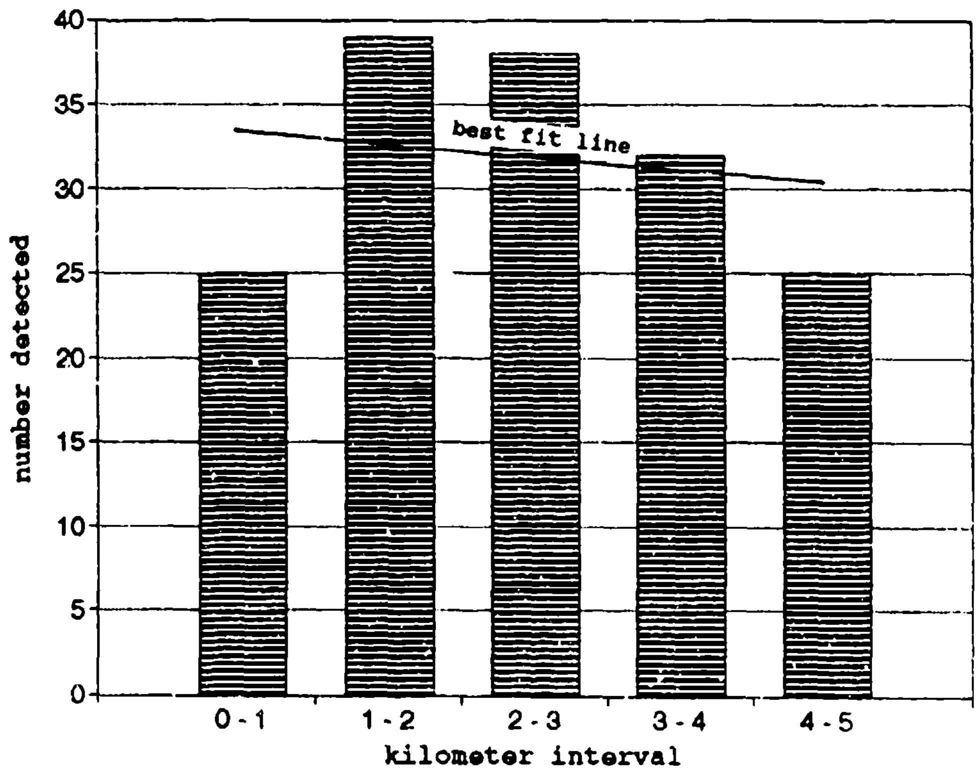




Figure 5.11 - Histogram for Case 8 using Detection Function C, after 20 program runs, n = 84 (above) and 34 program runs, n = 159.

## 5.4 Conclusions and Recommendations

The program test results indicate that the program is running as intended. The population density found using the program is very close to the actual population density input. The program does produce different results for different input whether the acoustic range of the hydrophone array, the speed of the towing ship or the detection function is changed. In addition, the program was intended to simulate field conditions. As in real life the histograms show erratic and random results that only vaguely reflect that expected.

The results for the two dimensional mode indicate some discrepancies with the transect theory. Using a circle for a detection zone may be causing adverse effect on how the whales are detected. More whales are detected as the distance from the transect line increases even when the probability of detection is the same for all distances. This effect is also apparent in the three-dimensional mode.

There are similarities in the two-dimensional and three-dimensional modes, but there are also significant differences. The total number of whales detected in each case using the two-dimensional and three-dimensional modes are nearly the same. For instance, in the two-dimensional Case 1, 94 whales were detected in 20 program runs and in the three-dimensional Case 6, 89 whales were detected

in 20 program runs.

The use of a circle and semisphere for detection zones may be causing a type of spreading effect in the results. More whales are detected as the distance from the transect line increases because the area or volume in which whales are detected increases proportionally with increasing range. This may indicate that the geometrical shpape of the type of detection zone used in line transect theory does effect the resulting density estimates.

The present number of program runs completed is not enough to make the results conclusive. Enough program runs should be completed for each test case so that the number of whales detected is greater than 400. This will provide a high confidence level and a small limit of error. Unfortunately as the program is currently designed, running the program enough times to get 400 whales detected for any test case will take a considerable amount of time. One program run with the input for Case 1 requires approximately 1 hour on a 486, 33 MHz personnel computer. The time for one program run for the three-dimensional mode with the same input takes 1.5 hours. Simply increasing the speed of the towing ship will not produce the desired results since as shown in Case 2 this would be unrealistic input and it decreases the number of whales that are detected.

## Appendix A - Notation

A - area
a - unknown parameter
D - density
DI - directivity index, in dB
DT - detection threshold, in dB
dB - decibel
E(n) - expected number of animals
g(x) - detection function
Hz - cycles per second
I - intensity
L - length of transect line
$\lambda$ - wavelength
N - number of animals in an sampled area
NL - noise level, in dB
P - probability
$P_w$ - power
p - pressure
RL - reverberation level, in dB
r - radius
SL - source level, in dB
TL - transmission loss, in dB
TS - target strength, in dB
t - time
x - horizontal distance
w - width
y - distance
z - depth distance

## Appendix B - References

Albers, V. M. (1969) Underwater Acoustics Instrumentation. Instrument Society of America, Pittsburgh, PA.

Anderson, G. and Evans, W. (1980) A Brief Description of a Special Purpose Towed Acoustic Array. Hubbs/Sea World Research Institute Technical Report Number 80-126.

Berg, R. E. and Stork, D. G. (1982) The Physics of Sound, Prentice-Hall, Inc., Englewood Cliffs, N.J.

Burnham, K. P., Anderson, D. R., and Laake, J. L. (1980) Estimation of Density from Line Transect Sampling of Biological Populations. Wildlife Monographs 72, 202 pp. A publication of the Journal of Wildlife Management.

Caruthers, J. W. (1977) Fundamentals of Marine Acoustics, Elsevier Scientific Publishing Company, Amsterdam, The Netherlands.

Coates, R. F. W. (1989) Underwater Acoustic Systems. Halsted Press, John Wiley & Sons, Inc., New York.

Cooke, J. G. (1985) Notes on the estimation of whale density from line transects. Report of the International Whaling Commission 35,319-323.

Cousteau, J. Y. and Diole P. (1972) The Whale: Mighty Monarch of the Sea, A & W Publishers, Inc., New York.

Cousteau, J. Y. (1988) Whales, Harry N. Abrams, Inc., New York.

Haines, G. (1974) Sound Underwater, Crane Russak & Company, Inc., New York.

Harrison, R. and Bryden, M. M., editors (1988) Whales, Dolphins, and Porpoises. Intercontinental Publishing Corporation, Hong Kong.

Jefferson, T. A., Leatherwood, S., Shoda, L. K. M., and Pitman, R. L. (1992) Marine Mammals of the Gulf of Mexico, A Field Guide for Aerial and Ship Board Observers, Texas A&M University Publishing Center, College Station, TX.

Knudsen, V. O., Alford, R. S., and Emling, J. W. (1948) Underwater Ambient Noise. In: Underwater Sound. V. M. Albers, editor, Dowden, Hutchinson & Ross, Inc., Stroudsburg, PA.

Stefanick, T. (1987) Strategic Antisubmarine Warfare and Naval Strategy, Institute for Defense & Disarmament Studies.

Urick, R. J. (1983) Principles of Underwater Sound, McGraw-Hill, Inc.

Watkins, W. A. (1977a) Acoustic behavior of sperm whales. Oceanus, **20**, 50-58.

Watkins, W. A. (1977b) Sperm whale codas. Journal of the Acoustical Society of America, **62**, 1485-2490.

Watkins, W. A., Moore, K. E., and Tyack, P. (1985) Sperm whale acoustic behaviors in the Southeast Caribbean. Cetology, **49**, 1-15.

Watkins, W. A. and Schevill, W. E. (1975) Sperm whale (physeter catodon) react to pingers. Deep-Sea Research , **22**, 123-129.

Watlington, F. (1979) How to Build & Use Low-Cost Hydrophones, Tab Books, Blue Ridge Summit, PA.

Wenz, G. M. (1962) Acoustic Ambient Noise in the Ocean: Spectra and Sources. In: Underwater Sound, V. M. Albers, editor, Dowden, Hutchinson & Ross, Inc., Stroudsburg, PA.

# Appendix C - Program code and instructions

A.1 This appendix contains the program code and the instructions on how to use "Line Transect Simulation - One", the program code for "Histogram converter" is also included. An example of program output is also provided.

## A.2 Instructions: How to use "Line Transect Simulation - One"

The program will run on any IBM or compatible personal computer with DOS 5.1, under QBASIC. The program is not compiled.

The program name is "WHALER.BAS".

After retrieving the file within the QBASIC environment press Alt R or use the mouse to pick "RUN" from the top menu. Pick "Start".

The program will begin with a cleared computer screen, a welcoming line and the request for user input on whether to use the graphic version or nongraphic version will appear:

---

```
"WELCOME
 TO THE LINE TRANSECT OCEAN SIMULATION PROGRAM"




"do you want the graphics program
 or no graphics program?"
```

---

Your reply must be a "Y" or "N", depending on your choice.

The screen will clear again and there will be additional questions:

---

```
"How many whales do you want?"

"What is the range of your acoustical equipment?"

"What is the speed of the boat in
 knots (1 nautical mile/hr = 1.852 kph = 1.151 mph)?"

"Do you want 2-D or 3-D (pick 2 or 3)?"
```

"Please input the file name of the detection function including the directory, path and extension."

"Is the file name correct (Y or N)?"

---

The first question, "How many whales do you want?", requires the total number of whales you want to occupy the ocean. The number may be between 0 and 501, however, as the number of whales increases the time required to run the program also increases significantly. A recommended amount of whales is 250 since this gives a density of whales of 0.001 whales/km$^2$.

The next question concerns the range of the hydrophone array being towed by the ship. The range is the radius of a circle (2-D) or hemisphere (3-D) that surrounds the ship. During the program run if a whale is located at a distance from the ship less than the range, then the whale will have a chance of being detected. Whales outside the range are not considered for possible detection.

The speed of the boat must be in knots although this value is converted into kilometers per minute while the program runs. The time unit used within the program is minutes and the unit of measure is the kilometer so a conversion from knots is required. The program is designed to simulate real conditions, so a speed of between 5 to 10 knots is recommended to portray the program as a proper simulation. Unfortunately the speed of the ship controls the speed of the program. The slower the speed of the towing ship, the slower the program runs since the ship movements are changed every program minute, but the distance traveled each program run remains the same. If a speed of 7 knots is used in the two-dimensional environment, then the time for one program run will be approximately 1 1/2 hours on a 486 personnel computer. If the three-dimensional mode is used the time required is nearly double at 3 hours.

You must choose either two or three dimensions for the program run. The two dimension environment excludes all depth measurements from the calculations. There is one exception to this. If the whale is within 5 meters of the surface it can n[illegible] be detected whether the program is running in three dimension [illegible] two dimensions. The three dimension environment includes [illegible]pth coordinate in the calculations and in the output.

An ASCII file of the detection function is required to run the program. The file must contain only numbers, real or integer, one per line. Each number is a probability corresponding the perpendicular distance from the transect line. The first number must be the probability for a distance of zero from the transect. The last number in the list is the probability of detection at the very edge of the acoustical equipment range. The program is

designed so that the detection function file may have 101 probabilities. An example would be:

1
0.99
0.98
0.97
.
.
.
0.02
0.01

The first probability value on the list corresponds to g(x) when x = 0. The probabilities then decrease as the distance x increases. Examples of detection functions are provided in Appendix A.

After these questions the screen will clear again and there will be questions concerning the program output. These are:

---

"Do you want the program output to go
to the Printer, File or Neither (P/F/N)?"

(If the you type "F" then . . .)

"Please input the file name including the directory,
path and extension."

"Is the file name correct?"

"Do you want a separate output file for the number of whales detected during each program run (Y or N)?"

(If "Y" then . . . )

"Please input the file name including the directory,
path and extension."

"Is the file name correct?"

"Do you want a separate file for the values of the perpendicular distance between detected whale and transect line (Y or N)?"

(If "Y" then . . .)

"Please input the file name including the directory,
path and extension."

"Is the file name correct?"

"If you would like the program to run more than once with the same input, please input the number of runs you would like: "

---

You can get up to three output files from the program run. Each file may already exist with data contained in it. The program will only append the file with the new data. It will not affect the existing data in the file.

All output files are in ASCII code. The first file name input will contain all the information shown on the screen during the program run, except the graphics, at the completion of the program. An example of this file output is shown in Chapter 5.

The next file name requested will contain only a listing with two columns. The first column is the total number of whales and the second column is the number of whales detected during a run. Each line represents a new program run. If you choose to run the program three times, but with different data so that your total number of whales is 100 for the first run, 200 for the second run and 250 for the last run, the file output may look like this:

```
100        4
200        6
250        5
```

The third output file will contain only the perpendicular distances between the detected whales and the transect line. This file is very easy to use in the program titled "Histogram Converter" included at the end of this appendix. Using the perpendicular distances the Histogram Converter provides ASCII output for use in creating a histogram. An output file containing perpendicular distances may look like this:

```
1.24
4.56
0.89
3.44
2.19
0.19
  .
  .
  .
```

The last question on the computer screen asks you how many times you want the program to run with the same input. You can press ENTER for one run or if you want more than run, input a number. For instance, if you want the program to run ten times with the same input just enter 10. At this the program will repeat 10 times with the same input, outputting to the same output files. At the end of each run all values go to zero or are reassigned to new

values.

Once you input the number of runs and press enter the program begins to run, the screen will clear. If you chose the graphics version you will get a two-dimensional view of the ocean, with whales and ship on the screen. In addition, information concerning the distance traveled by the ship is displayed, the time elapsed, your input data and a listing of all detected whales as the ship progresses across the screen.

If you did not chose the graphics program, the program should run faster, but the information displayed on the screen is limited to your relevant input data and the detected whales during a run and the number of program runs.

At the end of the program run or at the completion of all program runs specified by you, the total number of whales and the total number of whales detected will be tallied and shown on the screen.

The following question will appear:

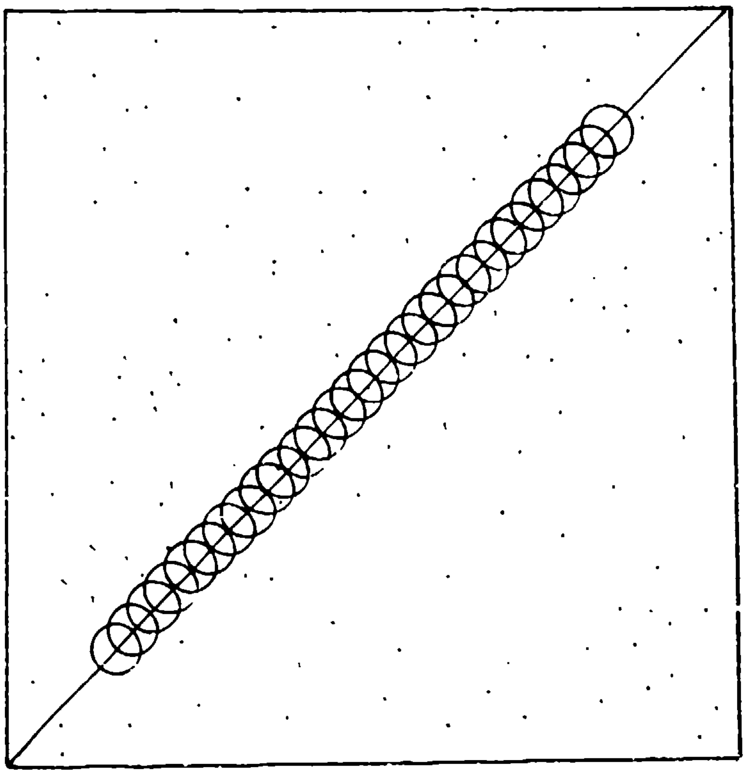
---

"Do you want to run the program again (Y/N)?"

---

If you press "Y" the program will start again from the beginning and new input will be required. If you push "N" the program will stop and you will return to the QBASIC environment.

Sample input and output screens are shown on the following pages. A sample of program output and the computer code follow.



LINE TRANSECT OCEAN SIMULATION
******************************

BOAT SPEED: 7 knots
RANGE: 5 km
DIMENSION: 2
TRANSECT LINE LENGTH: 500 km

| NO. | DISTANCE FROM SHIP | DISTANCE FROM LINE |
|---|---|---|
| 42 | 4.60 | 2.17 |
| 111 | 3.30 | 0.66 |
| 210 | 4.50 | 3.44 |
| 72 | 4.67 | 4.12 |
| 7 | 3.99 | 3.03 |

TIME: 11225
DISTANCE TRAVELED: 500 km

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 5

Do you want to run the program again (Y/N)?

DETECTION FUNCTION FILE NAME IS: C

************************************

LINE TRANSECT OCEAN SIMULATION DATA

BOAT SPEED: 7
RANGE: 5
DIMENSIONS: 3
LENGTH OF TRANSECT LINE: 500

- - - - - - - - - - - - - - - - - - - - - - -

RUN NUMBER: 1

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|
| 98 | 3.64 | 3.04 | 2.144 |
| 80 | 3.04 | 1.30 | 1.785 |
| 164 | 4.64 | 4.07 | 2.893 |
| 94 | 4.77 | 4.33 | 1.859 |
| 50 | 4.99 | 1.82 | 2.760 |

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 5

- - - - - - - - - - - - - - - - - - - - - - -

RUN NUMBER: 2

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|
| 35 | 4.91 | 2.06 | 2.760 |
| 114 | 2.51 | 2.48 | 1.897 |

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 2

- - - - - - - - - - - - - - - - - - - - - - -

RUN NUMBER: 3

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|
| 142 | 4.68 | 1.60 | 2.808 |
| 95 | 4.13 | 1.66 | 2.760 |
| 29 | 4.80 | 4.04 | 2.760 |
| 153 | 2.56 | 2.50 | 2.572 |
| 119 | 3.13 | 3.12 | 1.288 |

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 5

RUN NUMBER: 4

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|
| 235 | 4.48 | 2.10 | 1.547 |
| 195 | 4.85 | 1.86 | 1.306 |
| 5 | 3.25 | 1.53 | 2.251 |
| 2 | 4.94 | 0.89 | 2.760 |
| 197 | 4.24 | 4.23 | 2.792 |
| 245 | 4.83 | 2.51 | 2.792 |

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 6

- - - - - - - - - - - - - - - - - - - - - - -

RUN NUMBER: 5

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|
| 224 | 4.43 | 4.20 | 1.101 |
| 136 | 3.01 | 0.16 | 2.893 |
| 61 | 4.58 | 0.44 | 2.792 |
| 46 | 4.57 | 4.05 | 2.760 |
| 210 | 3.92 | 2.25 | 1.969 |
| 103 | 4.84 | 1.85 | 1.314 |

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 6

- - - - - - - - - - - - - - - - - - - - - - -

RUN NUMBER: 6

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|
| 56 | 4.88 | 0.76 | 2.465 |
| 198 | 3.22 | 1.51 | 2.893 |

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 2

RUN NUMBER: 7

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|
| 198 | 4.60 | 2.17 | 1.091 |
| 246 | 3.30 | 0.66 | 2.893 |
| 111 | 4.82 | 0.60 | 2.760 |
| 223 | 4.17 | 4.06 | 2.856 |
| 183 | 4.27 | 3.44 | 2.984 |
| 210 | 4.50 | 3.75 | 2.786 |

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 6

- - - - - - - - - - - - - - - - - - - - - - -

RUN NUMBER: 8

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|
| 94 | 4.95 | 2.94 | 2.760 |

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 1

- - - - - - - - - - - - - - - - - - - - - - -

RUN NUMBER: 9

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|
| 7 | 3.99 | 3.03 | 1.502 |
| 20 | 4.32 | 3.21 | 2.251 |
| 69 | 4.55 | 0.98 | 2.760 |
| 128 | 4.76 | 1.57 | 2.144 |
| 183 | 4.58 | 3.26 | 2.679 |
| 72 | 4.67 | 4.12 | 1.197 |
| 124 | 0.74 | 0.26 | 2.893 |

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 7

- - - - - - - - - - - - - - - - - - - - - - -

RUN NUMBER: 10

| WHALE NO. | DISTANCE TO SHIP | DISTANCE TO LINE | DEPTH |
|---|---|---|---|

TOTAL NUMBER OF WHALES: 250
TOTAL NUMBER OF WHALES DETECTED: 0

(Following is optional program output that shows the total number of whales and the number of whales detected per program run.)

| | |
|---|---|
| 250 | 5 |
| 250 | 2 |
| 250 | 5 |
| 250 | 6 |
| 250 | 6 |
| 250 | 2 |
| 250 | 6 |
| 250 | 1 |
| 250 | 7 |
| 250 | 0 |

(Following is optional program output that provides only the perpendicular distance between detected whales and the transect line.)

3.04
1.30
4.07
4.33
1.82
2.06
2.48
1.60
1.66
4.04
2.50
3.12
2.10
1.86
1.53
0.89
4.23
2.51
4.20
0.16
0.44
4.05
2.25
1.85
0.76
1.51
2.17
0.66
0.60
4.06
3.44
3.75
2.94
3.03
3.21
0.98
1.57
3.26
4.12
0.26

(from HISTO.BAS)

| | |
|---|---|
| .5 | 3 |
| 1 | 5 |
| 1.5 | 1 |
| 2 | 8 |
| 2.5 | 6 |
| 3 | 2 |
| 3.5 | 6 |
| 4 | 1 |
| 4.5 | 8 |
| 5 | 0 |

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'                    LINE TRANSECT SIMULATION - ONE
'
'
'This program simulates an ocean environment, with randomly swimming whales
'and a ship towing a linear hydrophone array.  The number of whales detected
'acoustically is compared to the total number of whales.
'
'The program is broken into two parts - graphics or nongraphics versions.
'There are 14 subprograms, each performing a different function within the
'main program.  The graphics version has 3 arrays, the nongraphic verion has
'only 2 arrays.
'
'Local variables are listed at the beginning of each subprogram.
'
'Global variables are:
'
'coordrec -    array record for whale coordinates used in creating graphics
'coord(n).x -  record element for whale x-coordinate
'coord(n).y -  record element for whale y-coordinate
'count% -      counter for number of whales detected per run
'dimension -   user specified either the second or third dimension
'f -           flag indicating the initial program and also elapsed program run time
'g(n) -        array that contains the detection function
'graph$ -      variable flag for graphic or nongraphic version
'infile$ -     variable for the name of the detection function file
'j -           counter for graphing whale movement
'no -          number of whales
'noofruns% -   counter for the number of runs with the same user input
'oput1$ -      flag for output, (f) means to file, (p) means to printer, (n) means none
'oput2$ -      name of file for program output, number of whales detected only
'oput3$ -      name of file for program output, perpendicular distances only
'range -       distance within whales can be detected
'runagain$ -   flag to RUN the program again
'runs% -       counter for the number of program runs with the same input
'uboat -       speed of the towing ship or boat
'whalerec -    array record of whale variables
'whale(n).x -  record element of whale x-coordinate at time t
'whale(n).y -  record element of whale y-coordinate at time t
'whale(n).z -  record element of whale z-coordinate at time t
'whale(n).d -  record element of whale diving depth
'whale(n).t -  record element of whale time (0 - 60 minutes)
'whale(n).s -  record element of whale diving (0) or submerging (1)
'whale(n).u -  record element of whale speed in x-direction
'whale(n).v -  record element of whale speed in y-direction
'whale(n).c -  record element showing if whale has been detected (1) or not (0)
'xboat -       x-coordinate of towing ship or boat at time t
'yboat -       y-coordinate of towing ship or boat at time t
'
'
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' Following are the declaration statements for all subprograms used.

DECLARE SUB OPENING (graph$)
DECLARE SUB GETINPUT (no, range, uboat, dimension, oput1$, oput2$, oput3$, runagain%, runs%, infile$)
DECLARE SUB DETECTWHALES (t, no, count%, nw%, xboat, yboat, range, dimension, oput1$, oput3$)
```

```
DECLARE SUB GRAPHWHALES (no, F, j)
DECLARE SUB GRAPHBOAT (xboat, yboat, F)
DECLARE SUB MOVEBOAT (xboat, yboat, uboat, t, F, runs%, noofruns%)
DECLARE SUB MOVEMENT (t, no, F)
DECLARE SUB CREATEWHALES (no)
DECLARE SUB SETUPFORRUN (uboat, range, dimension, runs%, noofruns%)
DECLARE SUB SETUPOUTPUT (range, uboat, dimension, oput1$, infile$, noofruns%)
DECLARE SUB TALLY (no, count%, nw%, runagain$, oput1$, oput2$, runs%, noofruns%)

' These subprograms are used in the non-graphics version:

DECLARE SUB NDETECTWHALES (t, no, count%, xboat, yboat, range, dimension, oput1$, oput3$)
DECLARE SUB NSETUPFORRUN (uboat, range, dimension, runs%, noofruns%)
DECLARE SUB NTALLY (no, count%, runagain$, oput1$, oput2$, runs%, noofruns%)

' The following TYPE block sets up a record for each individual whale.
' This TYPE block is used in both the graphic and non-graphics versions.

TYPE whalerec
  x AS SINGLE
  y AS SINGLE
  z AS SINGLE
  u AS SINGLE
  v AS SINGLE
  d AS SINGLE
  s AS SINGLE
  t AS INTEGER
  c AS INTEGER
END TYPE

' Following DIM statements dimension the arrays which hold the records of
' each individual whale set up above by the TYPE block and the detection
' function file input by the user. The dynamic statement makes the arrays
' dynamic instead of static arrays.

'$DYNAMIC
DIM SHARED whale(500) AS whalerec
DIM SHARED g(100)

' Following calls the opening subprogram to begin the program. The next call
' clears the screen and requests user input.

CALL OPENING(graph$)
CALL GETINPUT(no, range, uboat, dimension, oput1$, oput2$, oput3$, runagain%, runs%, infile$)

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Following IF determines if the user wants the graphics or non-graphics
' version of the program.

IF graph$ = "Y" THEN

' The following TYPE block sets up a record used for the graphics animation.

   TYPE coordrec
      x1 AS SINGLE
      x2 AS SINGLE
      x3 AS SINGLE
      x4 AS SINGLE
      x5 AS SINGLE
      x6 AS SINGLE
```

```
      x7 AS SINGLE
      x8 AS SINGLE
      y1 AS SINGLE
      y2 AS SINGLE
      y3 AS SINGLE
      y4 AS SINGLE
      y5 AS SINGLE
      y6 AS SINGLE
      y7 AS SINGLE
      y8 AS SINGLE
   END TYPE

' Following are the dimension statement for graphics arrays.

   DIM SHARED coord(500) AS coordrec

   WIDTH 80, 50

   DO
      noofruns% = noofruns% + 1
      CALL SETUPFORRUN(uboat, range, dimension, runs%, noofruns%)
      CALL SETUPOUTPUT(range, uboat, dimension, oputl$, infile$, noofruns%)

      RANDOMIZE TIMER

      t = 1
      F = 0
      j = 0
      xboat = 73
      yboat = 73
      count% = 0

      CALL CREATEWHALES(no)

      DO
         CALL MOVEBOAT(xboat, yboat, uboat, t, F, runs%, noofruns%)
         CALL GRAPHBOAT(xboat, yboat, F)
         CALL MOVEMENT(t, no, F)
         CALL GRAPHWHALES(no, F, j)
         CALL DETECTWHALES(t, no, count%, nw%, xboat, yboat, range, dimension, oputl$, oput3$)
         IF count% > 14 THEN count% = 0
         F = F + t
         j = j + 1
         IF j > 7 THEN j = 0
      LOOP UNTIL xboat >= 500

      CALL TALLY(no, count%, nw%, runagain$, oputl$, oput2$, runs%, noofruns%)

   LOOP UNTIL noofruns% = runs%

   IF runagain$ = "Y" THEN RUN

   CLOSE

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Following ELSEIF is used if the user does not want graphics.

ELSEIF graph$ = "N" THEN
```

```
  DO

    noofruns% = noofruns% + 1

    CALL MSETUPFORRUN(uboat, range, dimension, runs%, noofruns%)
    CALL SETUPOUTPUT(range, uboat, dimension, oputl$, infile$, noofruns%)

    RANDOMIZE TIMER

    t = 1
    F = 0
    xboat = 73
    yboat = 73
    count% = 0

    CALL CREATEWHALES(no)

    DO
       CALL MOVEBOAT(xboat, yboat, uboat, t, F, runs%, noofruns%)
       CALL MOVEMENT(t, no, F)
       CALL MDETECTWHALES(t, no, count%, xboat, yboat, range, dimension, oputl$, oput3$)
       F = F + t
    LOOP UNTIL xboat >= 428

    CALL NTALLY(no, count%, runagain$, oputl$, oput2$, runs%, noofruns%)

  LOOP UNTIL noofruns% = runs%

  IF runagain$ = "Y" THEN RUN

  CLOSE

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
END IF

END


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB CREATEWHALES (no)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'This subprogram randomly places a chosen number of whales (no) into a square
'"ocean". The x, y and z (depth) of the whale is random. This sub also
'randomly determines an initial horizontal velocity for the whale, whether the
'whale is at the surface, at its maximum "dive depth" or ascending or
'descending. All whale info for each whale is stored in a type file called
'whalerec.
'
'Locale variables:
'
'n - counter
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' Following DO LOOP gives each whale record a random location or x,y,z
' coordinates, a diving depth, a flag (s) to determine whether the whale
' is submerging or surfacing in its diving cycle, and a horizontal velocity
' vector with components u and v. The depth coordinate and diving depth
' coordinate are in meters in this subprogram.
```

```
DO
   whale(n).x = (RND * 500)
   whale(n).y = (RND * 500)

   whale(n).z = 3000 - INT(RND * 2000)
   whale(n).d = 1000 + INT(RND * 1000)

   whale(n).s = INT(RND * 2)

' Following creates a random velocity for each whale in kilometers
' per minute.

   whale(n).u = (4 - (8 * RND)) / 10
   whale(n).v = (4 - (8 * RND)) / 10

' Following IF THEN ensures that in case the whale's depth coordinate, z,
' is randomly generated to be lower than the whale's diving depth (the
' deepest the whale can go), then the whale's depth coordinate, z, is
' reassigned to the whale's diving depth, d.

   IF whale(n).d > whale(n).z THEN whale(n).z = whale(n).d

   n = n + 1

LOOP UNTIL n = no

END SUB

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB DETECTWHALES (t, no, count%, nw%, xboat, yboat, range, dimension, oput1$, oput3$)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Subfunction LOCATINGWHALES locates whales within the acoustic range specified
'by the user.  The detection function subprogram is used to determine if the
'whale located within the acoustic range has the probability of being detect-
'ed.  If the probability is no, the whale is not considered detected even
'though the whale is within the acoustic range.  Also, if the whale is within
'5 meters of the surface he can not be detected acoustically since he will
'not be making any noise.
'
'Local variables:
'
'angle2 - angle between transect line and the straight line between whale and boat
'chance - random number used as probability to compare against P
'm -      counter for DO LOOP
'mb -     slope of transect line
'mw -     slope of staight line between whale and boat
'P  -     probability from the detection function, g(x)
'p2 -     2-D perpendicular distance between whale and transect line
'p3 -     3-D perpendicular distance between whale and transect line
'r2 -     2-D straight line distance between whale and boat
'r3 -     3-D straight line distance between whale and boat
'ratio% - percentage, obtained by perpendicular distance from the whale to the
'         transect line divided by the range, used to get a P from the array g(x)
'xboat -  distance towing ship or boat has traveled
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' Following calculates the distance traveled along the transect line.
```

```
rboat = SQR(xboat ^ 2 + yboat ^ 2)

' Following prints the distance traveled on the screen.

LOCATE 25, 23
PRINT USING "###.## &"; rboat - 103.24; "km"

IF rboat > 103 AND rboat < 605 THEN

' The following draws a circle on the program graphic display that
' represents the acoustic range.

   CIRCLE (xboat, yboat), range, 2

' The following DO runs each of the whales in the ocean through
' a segment of code.

DO

' Following calculates the two-dimensional, horizontal distance
' between the whale and the towing ship.

      r2 = SQR((whale(m).x - xboat) ^ 2 + (whale(m).y - yboat) ^ 2)

' Following calculates the two-dimensional, horizontal distance
' between the whale and the transect line.


      SELECT CASE dimension

         CASE IS = 2

            IF r2 <= range AND whale(m).c = 0 AND whale(m).z < 2.5[illegible] THEN

               IF (whale(m).x - xboat) = 0 THEN
                  mw = 0
               ELSE
                  mw = (whale(m).y - yboat) / (whale(m).x - xboat)
               END IF

               mb = 1
               angle2 = ABS(ATN(mb) - ATN(mw))
               p2 = r2 * SIN(angle2)

               ratio% = (p2 / range * 100)
               P = g(ratio%)
               chance = (RND * 1)

               IF chance <= P THEN
                 whale(m).c = 1
                 count% = count% + 1
                 nw% = nw% + 1
                 BEEP
                 LOCATE (count% + 12), 50
                 PRINT USING " ###     ###.##      ###.##"; m; r2; p2
                 IF oput1$ = "F" THEN PRINT #2, USING " ###     ###.##      ###.##"; m; r2; p2
                 IF oput3$ = "Y" THEN PRINT #4, USING " ###.##"; p2
                 IF oput1$ = "P" THEN LPRINT USING " ###     ###.##      ###.## "; m; r2; p2
                 END IF
```

```
            END IF

          IF r2 > range THEN whale(m).c = 0

        CASE IS = 3

          r3 = SQR((whale(m).x - xboat) ^ 2 + (whale(m).y - yboat) ^ 2 + (3 - whale(m).z) ^ 2)

          IF r3 <= range AND whale(m).c = 0 AND whale(m).z < 2.995 THEN

            IF (whale(m).x - xboat) = 0 THEN
              mw = 0
            ELSE
              mw = (whale(m).y - yboat) / (whale(m).x - xboat)
            END IF

            mb = 1
            angle2 = ABS(ATN(mb) - ATN(mw))
            p2 = r2 * SIN(angle2)

            depth = 3 - whale(m).z
            p3 = SQR(p2 ^ 2 + depth ^ 2)

            ratio% = (p3 / range * 100)
            P = g(ratio%)
            chance = (RND * 1)

            IF chance <= P THEN
              whale(m).c = 1
              count% = count% + 1
              nw% = nw% + 1
              LOCATE (count% + 12), 49
              BEEP
              PRINT USING "###   ###.##   ###.##   #.###"; m; r3; p3; whale(m).z
              IF oput1$ = "F" THEN PRINT #2, USING " ###      ###.##      ###.##      #.###"; m; r3; p3; whale(m).z
              IF oput3$ = "Y" THEN PRINT #4, USING "###.##"; p3
              IF oput1$ = "P" THEN LPRINT USING " ###      ###.##      ###.##      #.###"; m; r3; p3; whale(m).z

              END IF

            END IF

          IF r3 > range THEN whale(m).c = 0

        END SELECT

m = m + 1
LOOP UNTIL m = no

END IF

END SUB

SUB GETINPUT (no, range, uboat, dimension, oput1$, oput2$, oput3$, runagain%, runs%, infile$)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Subfunction GETINPUT first clears the screen and takes the user's input,
'then it clears the screen again and sets up the screen for the program run.
'
'Local variables -
```

```
'
'checkin$ -     flag for checking file name of detection function
'checkout$ -    flag for checking first optional output file name
'checkit$ -     flag for checking second optional output file name
'checkyes$ -    flag for checking third optional output file name
'detectfile$ -  file name of second optional output file
'distancefile - file name of third optional output file
'm -            counter for DO LOOP to input detectio function into array g(m)
'outfile$ -     file name of first optional output file
'
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' Following IF THEN determines if the program is being run for the first time
' with new data.

IF runs% < 2 THEN

   CLS 0
   SCREEN 12

   DO
      LOCATE 3
      INPUT "How many whales do you want"; no$
      no = VAL(no$)
   LOOP UNTIL no > 0 OR LEN(no$) = 0

   DO
      LOCATE 5
      INPUT "What is the range of your acoustical equipment (in kilometers)"; range$
      range = VAL(range$)
      LOOP UNTIL range > 0 OR LEN(range$) = 0

   DO
      LOCATE 7
      PRINT "What is the speed of the boat in"
      LOCATE 8
      INPUT "knots (1 nautical mile/hr = 1.852 kph = 1.151 mph)"; uboat$
      uboat = VAL(uboat$)
   LOOP UNTIL uboat > 0 OR LEN(uboat$) = 0

   IF uboat = 0 OR LEN(uboat$) = 0 THEN uboat = 5

   knots = uboat

   DO
      LOCATE 10
      INPUT "Do you want 2-D or 3-D (pick 2 or 3)"; dimension
   LOOP UNTIL dimension = 2 OR dimension = 3

' Following DO inputs the detection function file name.

   DO
      DO
         LOCATE 12
         PRINT "Please input the file name of the detection function"
         LOCATE 13
         INPUT "including the directory, path and extension. ", infile$
      LOOP UNTIL LEN(infile$) > 0
```

```
      DO
         LOCATE 15
         INPUT "Is the file name correct (Y or N)? ", checkin$
      LOOP UNTIL LEN(checkin$) > 0

      IF LEFT$(UCASE$(checkin$), 1) <> "Y" THEN
         LOCATE 15
         PRINT "Please input the file name again.                        "
      END IF

   LOOP UNTIL LEFT$(UCASE$(checkin$), 1) = "Y"

   OPEN infile$ FOR INPUT AS #1

   PRINT " "

' Following Do Loop inputs detection function file into the array g(100).

   DO
      INPUT #1, g(m)
      m = m + 1
   LOOP UNTIL m > 100

   CLOSE #1

' Following Do Loop gets user input on where the program output
' should be sent, either in a seperate file ("F"), directly to the
' printer ("P") or neither one ("N"). If the user chooses "N" then the
' output will only be displayed on the computer screen.

   CLS 0

   DO
      LOCATE 3
      PRINT "Do you want program output to go"
      LOCATE 4
      INPUT "to the Printer, a File, or Neither (P/F/N)"; oputl$
   LOOP UNTIL LEFT$(UCASE$(oputl$), 1) = "P" OR LEFT$(UCASE$(oputl$), 1) = "F" OR LEFT$(UCASE$(oputl$), 1) = "N"

   oputl$ = LEFT$(UCASE$(oputl$), 1)

' The following If/Endif determines if the output is to go to a file and
' if the program has not been run immediately before. If the program has
' already run once, runagain = "Y", and the output file is already open.

   IF oputl$ = "F" AND CHR$(runagain%) <> "Y" THEN

      DO
         DO
            LOCATE 6
            PRINT "Please input the file name including the directory,"
            LOCATE 7
            INPUT "path and extension: ", outfile$
         LOOP UNTIL LEN(outfile$) > 0

         DO
            LOCATE 9
            INPUT "Is the file name correct (Y or N)? ", checkout$
         LOOP UNTIL LEN(checkout$) > 0
```

```
      IF LEFT$(UCASE$(checkout$), 1) <> "Y" THEN
         LOCATE 9
         PRINT "Please input file name again.                    "
      END IF

   LOOP UNTIL LEFT$(UCASE$(checkout$), 1) = "Y"

' Following gets the name of the file from the user for input of just the
' number of whales detected during each run.  If the user does not input a
' name then no file is created or opened for this information.

   DO
      LOCATE 11
      PRINT "Would you like the number of whales detected during each program"
      LOCATE 12
      INPUT "run to go to a separate file (Y/N)?  ", oput2$
   LOOP UNTIL LEFT$(UCASE$(oput2$), 1) = "Y" OR LEFT$(UCASE$(oput2$), 1) = "N"

   oput2$ = LEFT$(UCASE$(oput2$), 1)

   IF oput2$ = "Y" THEN

      DO

         DO
            LOCATE 13
            PRINT "Please input that file name with directory,"
            LOCATE 14
            INPUT "path and extension. ", detectfile$
         LOOP UNTIL LEN(detectfile$) > 0

         DO
            LOCATE 15
            INPUT "Is the file name correct (Y or N)? ", checkit$
         LOOP UNTIL LEFT$(UCASE$(checkit$), 1) = "Y" OR LEFT$(UCASE$(checkit$), 1) = "N"


         IF LEFT$(UCASE$(checkit$), 1) = "N" THEN
            LOCATE 15
            PRINT "Please input file name again.                    "
         END IF

      LOOP UNTIL LEFT$(UCASE$(checkit$), 1) = "Y"

   END IF

' Following DO LOOP determines if the user wants the perpendicular distance
' between the detected whale and the transect line to go to a separate file.
' If the user does, the file name is requested.  If the user does not then no
' file is opened.

   DO
      LOCATE 17
      PRINT "Would like the perpendicular distance between each detected"
      LOCATE 18
      INPUT "whale and the transect line to go to a separate file (Y/N)? ", oput3$
   LOOP UNTIL LEFT$(UCASE$(oput3$), 1) = "Y" OR LEFT$(UCASE$(oput3$), 1) = "N"

   oput3$ = LEFT$(UCASE$(oput3$), 1)
```

```
    IF oput3$ = "Y" THEN

      DO
        DO
          LOCATE 19
          PRINT "Please input that file name with the directory"
          LOCATE 20
          INPUT "path and extension.  ", distancefile$
        LOOP UNTIL LEN(distancefile$) > 0

        DO
          LOCATE 22
          INPUT "Is the file name correct (Y or N)? ", checkyes$
         LOOP UNTIL LEFT$(UCASE$(checkyes$), 1) = "Y" OR LEFT$(UCASE$(checkyes$), 1) = "N"

         IF LEFT$(UCASE$(checkyes$), 1) = "N" THEN
            LOCATE 22
            PRINT "Please input file name again.                       "
          END IF

      LOOP UNTIL LEFT$(UCASE$(checkyes$), 1) = "Y"

    END IF

' The following opens the user's files so that the program input may be
' added to it.

    OPEN outfile$ FOR APPEND AS #2
    IF LEN(detectfile$) > 0 THEN OPEN detectfile$ FOR APPEND AS #3
    IF LEN(distancefile$) > 0 THEN OPEN distancefile$ FOR APPEND AS #4

  END IF

' The following Do Loop determines how many times the user wants the
' program to run with the same input.

  DO
    LOCATE 24
    PRINT "If you would like the program to run more than once with the same"
    LOCATE 25
    INPUT "input, please input the number of runs you would like: ", runs$
    runs% = VAL(runs$)
    LOOP UNTIL runs% > 0 OR LEN(runs$) = 0

    IF LEN(runs$) = 0 THEN runs% = 1

END IF

END SUB

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB GRAPHBOAT (xboat, yboat, F)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'This subprogram plots the path of the towing-ship in the graphics version of
'the program.  This subprogram also prints the time elapsed on the graphics
'screen.  There are no local variables.
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
```

```
LINE (0, 0)-(xboat, yboat), 14

LOCATE 24, 10
PRINT F

END SUB

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB GRAPHWHALES (no, F, j)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'This subprogram is only used in the graphic version of the program.  The plot
'of each whales travel is shown on the graphic screen.  This is done by show-
'ing the location of the whale's position for the last eight minutes of prog-
'ram time.
'
'Local variables:
'
'n - counter for DO LOOP to run through all the whales up to "no"
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
DO

   IF whale(n).z = 3! THEN
      PSET (whale(n).x, whale(n).y), 15
   ELSE
      PSET (whale(n).x, whale(n).y), 7
   END IF

' Following IF and SELECT CASE are used in the graphical representation of
' whales on the screen.  The x and y coordinates of each whale for the
' preceeding 8 minutes is stored in an array and plotted by a point on the
' screen.

   SELECT CASE F

       CASE IS > 7

         IF j = 0 THEN
'           PSET (coord(n).x1, coord(n).y1), 1
           coord(n).x1 = whale(n).x
           coord(n).y1 = whale(n).y
        ELSEIF j = 1 THEN
'           PSET (coord(n).x2, coord(n).y2), 1
           coord(n).x2 = whale(n).x
           coord(n).y2 = whale(n).y
        ELSEIF j = 2 THEN
'           PSET (coord(n).x3, coord(n).y3), 1
           coord(n).x3 = whale(n).x
           coord(n).y3 = whale(n).y
        ELSEIF j = 3 THEN
'           PSET (coord(n).x4, coord(n).y4), 1
           coord(n).x4 = whale(n).x
           coord(n).y4 = whale(n).y
        ELSEIF j = 4 THEN
'           PSET (coord(n).x5, coord(n).y5), 1
           coord(n).x5 = whale(n).x
           coord(n).y5 = whale(n).y
        ELSEIF j = 5 THEN
'           PSET (coord(n).x6, coord(n).y6), 1
```

```
            coord(n).x6 = whale(n).x
            coord(n).y6 = whale(n).y
          ELSEIF j = 6 THEN
'              PSET (coord(n).x7, coord(n).y7), 1
            coord(n).x7 = whale(n).x
            coord(n).y7 = whale(n).y
          ELSEIF j = 7 THEN
'              PSET (coord(n).x8, coord(n).y8), 1
            coord(n).x8 = whale(n).x
            coord(n).y8 = whale(n).y
          END IF


          CASE IS = 0
            coord(n).x1 = whale(n).x
            coord(n).y1 = whale(n).y
          CASE IS = 1
            coord(n).x2 = whale(n).x
            coord(n).y2 = whale(n).y
          CASE IS = 2
            coord(n).x3 = whale(n).x
            coord(n).y3 = whale(n).y
          CASE IS = 3
            coord(n).x4 = whale(n).x
            coord(n).y4 = whale(n).y
          CASE IS = 4
            coord(n).x5 = whale(n).x
            coord(n).y5 = whale(n).y
          CASE IS = 5
            coord(n).x6 = whale(n).x
            coord(n).y6 = whale(n).y
          CASE IS = 6
            coord(n).x7 = whale(n).x
            coord(n).y7 = whale(n).y
          CASE IS = 7
            coord(n).x8 = whale(n).x
            coord(n).y8 = whale(n).y
        END SELECT

   n = n + 1

LOOP UNTIL n = no

END SUB

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB MOVEBOAT (xboat, yboat, uboat, t, F, runs%, noofruns%)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'Subprogram boat moves the boat diagonally across the ocean at a speed
'the user inputs.  The user inputs the boat speed in nautical miles per hour
'but the program units are kilometers per minute so a conversion is required.
'Boat also plots the boat and its path across the ocean on the graphic screen.
'
'There are no local variables.
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'  Following converts boat speed in knots into kilometers per minute.

IF F = 0 AND noofruns% = 1 THEN uboat = uboat * 1.852 / 60
```

```
' Following relocates the location of the boat every minute using the
' boat speed in kilometers per minute.

xboat = xboat + uboat * t
yboat = yboat + uboat * t

END SUB

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB MOVEMENT (t, no, F)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'This subprogram calculates the x,y,z coordinate movements for each whale for
'each increment of time, t.  The velocity of the whale in the horizontal
'direction (x and y) is randomized.  The vertical velocity is set by the
'variables, wdown and wup, which coorespond to the whales diving speed and the
'whales surfacing speed.
'
'In addition, the whales are not allowed to go beyond the boundaries of the
'"ocean".
'
'Local variables:
'
'wdown   - whale diving speed (set at 4 mph or 0.107 kpm)
'wup     - whale ascending speed (set at 6 mph or 0.160 kpm)
'n       - counter for each whale to run through the subprogram
'changev - variable used in randomly changing the whale's velocity
'
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
' Following sets the diving speed and ascending speed of the whales in
' meters per minute.  Calculations for depths are in meters and then
' converted to kilometers.

wdown = 107
wup = 16

' Following FOR loops all whales through a movement giving each whale
' a new location every minute.

DO

' When F = 0 the program has just begun, time elapsed is zero, and all whales
' are in their initial positions.  The following IF determines at what depth
' a whale is located and whether the whale is diving or surfacing, and based
' on this, assigns a whaletime unit between from 0 and 60 to that whale.

   IF F = 0 THEN
      IF whale(n).z = 3000 THEN
         whale(n).t = 0
      ELSEIF whale(n).z < 3000 AND whale(n).s = 0 THEN
         whale(n).t = 15 + (3000 - whale(n).z) / wdown
      ELSEIF whale(n).z < 3000 AND whale(n).s = 1 THEN
         IF (whale(n).d + 15 * wup) >= 3000 THEN
            whale(n).t = 45 + (whale(n).z - whale(n).d) / wup
         ELSE
            whale(n).z = 3000 - 15 * wup
            whale(n).d = whale(n).z
            whale(n).t = 46
         END IF
      ELSE whale(n).z = 3000 AND whale(n).t = 0
```

```
      END IF

' The ELSE of the above IF is used when F is no longer zero and movement of
' the boat and whales has begun.  The ELSE converts the depth to meters
' (later in this subprogram the depth is converted to kilometers) and
' adds another minute to the whaletime.

    ELSE
      whale(n).z = 1000 * whale(n).z
      IF whale(n).t < 60 THEN
         whale(n).t = whale(n).t + 1
      ELSE whale(n).t = 0
      END IF
    END IF

' Following IF determines what the depth and whaletime of the whale is, and
' using this information, moves the whale in the z-direction.

    IF whale(n).z >= 3000 AND whale(n).t <= 15 THEN
      whale(n).z = 3000
      whale(n).s = 0
    ELSEIF whale(n).z <= whale(n).d AND whale(n).t <= 45 THEN
      whale(n).z = whale(n).d
      whale(n).s = 1
    ELSEIF whale(n).s = 0 THEN
      IF whale(n).t <= 45 AND whale(n).z >= whale(n).d THEN
           whale(n).z = whale(n).z - wdown * t
      ELSE
           whale(n).z = whale(n).z + wup * t
           whale(n).s = 1
      END IF
    ELSEIF whale(n).s = 1 THEN
      IF whale(n).t > 45 AND whale(n).t < 60 AND whale(n).z < 3000 THEN
           whale(n).z = whale(n).z + wup * t
      ELSE whale(n).z = 3000
           whale(n).s = 0
      END IF
    ELSE whale(n).z = 3000
     whale(n).t = 0
    END IF

' Following IF prevents the whale from being moved up beyond the water
' surface and below the diving depth of the whale.

    IF whale(n).z > 3000 THEN
      whale(n).z = 3000
      whale(n).s = 0
    ELSEIF whale(n).z < whale(n).d THEN
      whale(n).z = whale(n).d
      whale(n).s = 1
    ELSE whale(n).z = whale(n).z
    END IF

' Following IF moves the whale in the x and y-directions.  If the whale is
' at the surface (z = 3000 m) then the whale only moves a third of the
' velocity it moves when underwater.  Also, the IF keeps the whales from
' moving out of the ocean boundaries of x = 0, y = 0, x = 500 and y = 500.

    IF whale(n).z >= 2998 THEN
```

```
      whale(n).y = whale(n).y + whale(n).v / 3 * t
      whale(n).x = whale(n).x + whale(n).u / 3 * t

      IF whale(n).y <= 0 OR whale(n).y >= 500 THEN
         whale(n).v = -whale(n).v
         whale(n).y = whale(n).y + whale(n).v / 3 * t
      END IF

      IF whale(n).x <= 0 OR whale(n).x >= 500 THEN
         whale(n).u = -whale(n).u
         whale(n).x = whale(n).x + whale(n).u / 3 * t
      END IF

   ELSE

      whale(n).y = whale(n).y + whale(n).v * t
      whale(n).x = whale(n).x + whale(n).u * t

      IF whale(n).y <= 0 OR whale(n).y >= 500 THEN
         whale(n).v = -whale(n).v
         whale(n).y = whale(n).y + whale(n).v * t
      END IF

      IF whale(n).x <= 0 OR whale(n).x >= 500 THEN
         whale(n).u = -whale(n).u
         whale(n).x = whale(n).x + whale(n).u * t
      END IF

   END IF

' Following converts the whale depth from meters to kilometers (depth is
' converted from kilometers to meters in an IF-ELSE statement above).

   whale(n).z = whale(n).z / 1000

' Following generates a random number used in the IF statement to decide
' whether to change the whale's horizontal (x and y) velocities.  This
' ensures the whales are always moving at a different speed.

   changev = (RND * 10)
   IF changev < 2 THEN
      whale(n).u = (4 - (8 * RND)) / 10
      whale(n).v = (4 - (8 * RND)) / 10
   END IF

   n = n + 1

LOOP UNTIL n = no

END SUB

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB NDETECTWHALES (t, no, count%, xboat, yboat, range, dimension, oput1$, oput3$)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

'Subfunction LOCATINGWHALES locates whales within the acoustic range specified
'by the user.  The detection function subprogram is used to determine if the
'whale located within the acoustic range has the probability of being detect-
'ed.  If the probability is no, the whale is not considered detected even
'though the whale is within the acoustic range.  Also, if the whale is within
'5 meters of the surface he can not be detected acoustically since he will
```

```
'not be making any noise.
'
'Local variables:
'
'angle2 - angle between transect line and straight line between whale and boat
'chance - randomly generated probability to be compare to P
'depth -  vertical distance between the surface and the whale
'm -      counter for DO LOOP
'mb -     slope of transect line
'mw -     slope of straight line between whale and boat
'P  -     probability from the detection function, g(x)
'p2 -     2-D perpendicular distance from whale to transect line
'p3 -     3-D perpendicular distance from whale to transect line
'r2 -     2-D distance between whale and boat
'r3 -     3-D distance between whale and boat
'ratio% - percentage derived from the perpendicular distance divided by the range
'rboat -  distance boat has traveled

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' Following calculates the distance traveled along the transect line.

rboat = SQR(xboat ^ 2 + yboat ^ 2)

' Following IF allows whale detection for only a distance of 500 km.

IF rboat > 103 AND rboat < 605 THEN

' The following DO runs each of the whales in the ocean through
' a segment of code.

   DO

' Following calculates the two-dimensional, horizontal distance
' between the whale and the towing ship.

      r2 = SQR((whale(m).x - xboat) ^ 2 + (whale(m).y - yboat) ^ 2)

' Following calculates the two-dimensional, horizontal distance
' between the whale and the transect line.


' Following SELECT CASE has a case for two-dimensions or three-dimensions.

      SELECT CASE dimension

         CASE IS = 2

            IF r2 <= range AND whale(m).c = 0 AND whale(m).z < 2.995 THEN

               IF (whale(m).x - xboat) = 0 THEN
                  mw = 0
               ELSE
                  mw = (whale(m).y - yboat) / (whale(m).x - xboat)
               END IF

               mb = 1
               angle2 = ABS(ATN(mb) - ATN(mw))
               p2 = r2 * SIN(angle2)
```

```
          ratio% = (p2 / range * 100)
          P = g(ratio%)
          chance = (RND * 1)

          IF chance <= P THEN
            whale(m).c = 1
            count% = count% + 1
            PRINT USING "   ###              ###.##                  ###.##"; m; r2; p2
            IF oput1$ = "P" THEN PRINT #2, USING "   ###              ###.##                  ###.##"; m; r2; p2
            IF oput3$ = "Y" THEN PRINT #4, USING "###.##"; p2
            IF oput1$ = "P" THEN LPRINT USING "   ###              ###.##                  ###.## "; m; r2; p2
            END IF

          END IF

        IF r2 > range THEN whale(m).c = 0

      CASE IS = 3

        r3 = SQR((whale(m).x - xboat) ^ 2 + (whale(m).y - yboat) ^ 2 + (3 - whale(m).z) ^ 2)

        IF r3 <= range AND whale(m).c = 0 AND whale(m).z < 2.995 THEN

          IF (whale(m).x - xboat) = 0 THEN
             mw = 0
          ELSE
             mw = (whale(m).y - yboat) / (whale(m).x - xboat)
          END IF

          mb = 1
          angle2 = ABS(ATN(mb) - ATN(mw))
          p2 = r2 * SIN(angle2)

          depth = 3 - whale(m).z
          p3 = SQR(p2 ^ 2 + depth ^ 2)

          ratio% = (p3 / range * 100)
          P = g(ratio%)
          chance = (RND * 1)

          IF chance <= P THEN
            whale(m).c = 1
            count% = count% + 1
            PRINT USING "   ###              ###.##                  ###.##                  #.###"; m; r3; p3; whale(m).z
            IF oput1$ = "P" THEN PRINT #2, USING "   ###              ###.##                  ###.##                  #.###"; m; r3;
            IF oput3$ = "Y" THEN PRINT #4, USING "###.##"; p3
            IF oput1$ = "P" THEN LPRINT USING "     ###              ###.##                  ###.##                  #.###"; m; r3;

            END IF

          END IF

        IF r3 > range THEN whale(m).c = 0

      END SELECT

   m = m + 1
   LOOP UNTIL m = no

 END IF
```

```
END SUB

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB NSETUPFORRUN (uboat, range, dimension, runs%, noofruns%)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'This subprogram sets up the screen for the program output if the nongraphic
'version of the program is chosen.  There are no local variables.
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' Following clears for program output if it is the first run.

IF noofruns% = 1 THEN

   CLS 0

   PRINT " "
   PRINT " LINE TRANSECT OCEAN SIMULATION"
   PRINT "**********************************"
   PRINT " "
   PRINT USING "& ### &"; "BOAT SPEED:"; uboat; "knots"
   PRINT USING "& ### &"; "RANGE:     "; range; "km"
   PRINT USING "& #"; "DIMENSION:  "; dimension
   PRINT "TRANSECT LINE LENGTH:  500 km"
   PRINT " "
   PRINT USING "& ###"; "RUN NUMBER:  "; noofruns%

   SELECT CASE dimension
     CASE 2
        PRINT " "
        PRINT "WHALE NO.    DISTANCE TO SHIP     DISTANCE TO LINE"
        PRINT " "
     CASE 3
        PRINT " "
        PRINT "WHALE NO.    DISTANCE TO SHIP     DISTANCE TO LINE         Z "
        PRINT " "
     END SELECT

ELSE
   PRINT " "
   PRINT USING "& ###"; "RUN NUMBEk:  "; noofruns%
   PRINT " "
END IF


END SUB

SUB NTALLY (no, count%, runagain$, oput1$, oput2$, runs%, noofruns%)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'This subprogram displays the number of whales and the number of whales
'detected per program run at the end of the program run.  Also the user
'is asked if the program should be run again.  There are no local variables.
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
PRINT " "
PRINT USING "& ###"; "TOTAL NUMBER OF WHALES:            "; no
PRINT USING "& ###"; "TOTAL NUMBER OF WHALES DETECTED:  "; count%
```

```
PRINT "   "

IF oput1$ = "F" THEN

   PRINT #2, " "
   PRINT #2, USING "& ###"; "TOTAL NUMBER OF WHALES:          "; no
   PRINT #2, USING "& ###"; "TOTAL NUMBER OF WHALES DETECTED:  "; count%
   PRINT #2, "  "

END IF

IF oput2$ = "Y" THEN PRINT #3, USING "###    ###"; no; count%

IF oput1$ = "P" THEN

   LPRINT , " "
   LPRINT USING "& ###"; "TOTAL NUMBER OF WHALES:          "; no
   LPRINT USING "& ###"; "TOTAL NUMBER OF WHALES DETECTED:  "; count%

END IF

IF noofruns% = runs% THEN

   DO
      PRINT "  "
      INPUT "Do you want to run the program again (Y/N)"; runagain$
      runagain$ = LEFT$(runagain$, 1)
      LOOP UNTIL UCASE$(runagain$) = "Y" OR UCASE$(runagain$) = "N"

   runagain$ = UCASE$(runagain$)

   END IF

END SUB

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB OPENING (graph$)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'This subprogram welcomes the user to the program and asks the user to decide
'whether the graphics or nongraphics version should be used.  There are no
'local variables.
'
'''''''''''''''''''''''''''''''''''' ''''''''''''''''''''''''''''''''''''''''''
CLS 0
SCREEN 12
VIEW (10, 10)-(629, 469), , 1
VIEW (40, 40)-(509, 449), , 1

LOCATE 14, 10
PRINT "WELCOME"
LOCATE 15, 10
PRINT "TO THE LINE TRANSECT OCEAN SIMULATION PROGRAM"

DO
   LOCATE 26, 20
   INPUT "do you want the graphics version (Y/N)? ", graph$
LOOP UNTIL LEFT$(UCASE$(graph$), 1) = "Y" OR LEFT$(UCASE$(graph$), 1) = "N"

graph$ = LEFT$(UCASE$(graph$), 1)
```

```
END SUB

SUB SETUPFORRUN (uboat, range, dimension, runs%, noofruns%)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'This program sets up the screen for the graphic version of the program.
'
'Local variables are:
'
'knots - speed of the boat in knots
'
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
CLS 0
SCREEN 12
VIEW (10, 10)-(360, 360), 1, 4
WINDOW (500, 500)-(0, 0)

IF noofruns% = 1 THEN knots = uboat

   LOCATE 2, 49
   PRINT " LINE TRANSECT OCEAN SIMULATION"
   LOCATE 3, 49
   PRINT "**********************************"

   LOCATE 5, 50
   PRINT USING "& ### &"; "BOAT SPEED:"; knots; "knots"

   LOCATE 6, 50
   PRINT USING "& ### &"; "RANGE:     "; range; "km"

   LOCATE 7, 50
   PRINT USING "& #"; "DIMENSION:   "; dimension

   LOCATE 8, 50
   PRINT "TRANSECT LINE LENGTH:  500 km"

   IF dimension = 2 THEN
      LOCATE 10, 50
      PRINT "       DISTANCE    DISTANCE "
      LOCATE 11, 50
      PRINT "NO.    FROM SHIP   FROM LINE"

   ELSE
      LOCATE 10, 50
      PRINT "      DIST     DIST    "
      LOCATE 11, 50
      PRINT "NO   TO SHIP  TO LINE    z"

   END IF

LOCATE 24
PRINT "TIME:"

LOCATE 25
PRINT "DISTANCE TRAVELED: "

IF runs% <> 1 THEN
   LOCATE 27
   PRINT USING "& ###"; "RUN NUMBER "; noofruns%
```

```
   END IF

END SUB

'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
SUB SETUPOUTPUT (range, uboat, dimension, oputl$, infile$, noofruns%)
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'Subprogram SETUPOUTPUT prints headings and initial program input to either
'a file specified by the user or the printer.
'
'There are no local variables.
'
'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


IF oputl$ = "F" AND noofruns% = 1 THEN

   PRINT #2, USING "& &"; "DETECTION FUNCTION FILE NAME IS:  "; infile$
   PRINT #2, "   "
   PRINT #2, "*************************************"
   PRINT #2, "LINE TRANSECT OCEAN SIMULATION DATA"
   PRINT #2, "   "
   PRINT #2, "BOAT SPEED:      ", uboat
   PRINT #2, "RANGE:           ", range
   PRINT #2, "DIMENSIONS:      ", dimension
   PRINT #2, "LENGTH OF TRANSECT LINE:     500 "
   PRINT #2, "  "
   PRINT #2, "- - - - - - - - - - - - - - - - - - - - - - - -"
   PRINT #2, "RUN NUMBER:      ", noofruns%
   PRINT #2, "  "
   PRINT #2, "WHALE           DISTANCE          DISTANCE     "

   IF dimension = 2 THEN
     PRINT #2, " NO.            TO SHIP           TO LINE"
     PRINT #2, "  "
   ELSE
     PRINT #2, " NO.            TO SHIP           TO LINE     DEPTH "
     PRINT #2, "  "
   END IF

END IF

IF oputl$ = "F" AND noofruns% > 1 THEN

   PRINT #2, "   "
   PRINT #2, "- - - - - - - - - - - - - - - - - - - - - - - -"
   PRINT #2, "RUN NUMBER:      ", noofruns%
   PRINT #2, "   "
   PRINT #2, "WHALE           DISTANCE          DISTANCE     "

   IF dimension = 2 THEN
     PRINT #2, " NO.            TO SHIP           TO LINE"
     PRINT #2, "  "
     ELSE
     PRINT #2, " NO.            TO SHIP           TO LINE     DEPTH "
     PRINT #2, "  "
   END IF

END IF
```

```
IF oputl$ = "P" AND noofruns% = 1 THEN

   LPRINT , "  "
   LPRINT , "**************************************"
   LPRINT , "LINE TRANSECT OCEAN SIMULATION DATA"
   LPRINT , "  "
   LPRINT , "BOAT SPEED:      ", knots
   LPRINT , "RANGE:           ", range
   LPRINT , "DIMENSIONS:      ", dimension
   LPRINT , "LENGTH OF TRANSECT LINE:  500",
   LPRINT , "  "
   LPRINT , "RUN NUMBER:      ", noofruns%
   LPRINT , "  "
   LPRINT , "WHALE        DISTANCE        DISTANCE"

   IF dimension = 2 THEN
      LPRINT , " NO.         TO SHIP         TO LINE "
      LPRINT ,
   ELSE
      LPRINT , " NO.         TO SHIP         TO LINE     DEPTH"
      LPRINT , "  "
   END IF

END IF

IF oputl$ = "P" AND noofruns% > 1 THEN

   LPRINT , "  "
   LPRINT , "RUN NUMBER:      ", noofruns%
   LPRINT , "  "

   IF dimension = 2 THEN
      LPRINT , " NO.         TO SHIP         TO LINE "
      LPRINT ,
   ELSE
      LPRINT , " NO.         TO SHIP         TO LINE     DEPTH"
      LPRINT , "  "
   END IF

END IF

END SUB

SUB TALLY (no, count%, nw%, runagain$, oputl$, oput2$, runs%, noofruns%)
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'
'This subprogram prints up the number of whales and the number of whales
'detected per program run at the end of each program run.  If also asks the
'user if the program should run again.  There are no local variables.
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

IF runs% = 1 THEN
   LOCATE 27
   PRINT "TOTAL NUMBER OF WHALES:           "; no
   PRINT "TOTAL NUMBER OF WHALES DETECTED:  "; nw%
END IF

IF oputl$ = "P" THEN
```

```
   PRINT #2, " "
   PRINT #2, "TOTAL NUMBER OF WHALES:           ", no
   PRINT #2, "TOTAL NUMBER OF WHALES DETECTED:  ", nw%
END IF

IF oput2$ = "Y" THEN PRINT #3, USING "###   ###"; no; nw%

IF oput1$ = "P" THEN
   LPRINT , " "
   LPRINT , "TOTAL NUMBER OF WHALES:         ", no
   LPRINT , "TOTAL NUMBER OF WHALES DETECTED:", nw%
END IF

IF noofruns% = runs% THEN

   DO
      LOCATE 29
      INPUT "Do you want to run the program again (Y/N)"; ragain$
      runagain$ = LEFT$(ragain$, 1)
   LOOP UNTIL UCASE$(runagain$) = "Y" OR UCASE$(runagain$) = "N"

   runagain$ = UCASE$(runagain$)

END IF

END SUB
```

```
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
'                          HISTOGRAM CONVERTER
'
'This program will take an ASCII file of a listing of values and output the
'number of values per class interval.  The class interval is specified by the
'user.
'The output data may be used to create a histogram of the original ASCII file
'data.
'
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

' Following TYPE sets up a record for the class interval and the number of
' values in the class interval.

TYPE historec
   classinterval AS SINGLE
   number AS INTEGER
END TYPE

CLS 0

PRINT "************************"
PRINT "CONVERT DATA TO HISTOGRAM DATA"
PRINT "  "

INPUT "What is the highest number of the range (example: range = 5 km)? ", hrange
INPUT "What is the lowest number of the range (example: 0 km)? ", lrange
INPUT "What is the class interval (example 0-0.5, 0.5-1.0 interval = 0.5)? ", interval
PRINT "  "
INPUT "What is the name of the ASCII file your data is in? ", finame$
INPUT "What is the name of the ASCII file for program output? ", outname$

OPEN finame$ FOR INPUT AS #1

length% = LOF(1) / 2

DIM d(length%)

m% = 0

DO UNTIL EOF(1)
   INPUT #1, d(m%)
   PRINT d(m%)
   m% = m% + 1
LOOP

CLOSE #1

range = hrange - lrange
increment% = range / interval
t = interval

PRINT "  "

DIM h(increment%) AS historec

DO
   h(n%).classinterval = lrange + t

   t = t + interval
```

```
   n% = n% + 1

LOOP UNTIL n% = increment%

PRINT " "

k% = m% - 1

FOR s% = 0 TO k%

   t = interval
   r% = 0

   DO

      IF d(s%) <= lrange + t THEN
         h(r%).number = h(r%).number + 1
         r% = increment%
      ELSE
         r% = r% + 1
         t = t + interval
      END IF

   LOOP UNTIL r% = increment%

NEXT s%

PRINT " "
PRINT "Number of whales located:  ", m%
PRINT " "

OPEN outname$ FOR APPEND AS #2

DO
   PRINT #2, h(q).classinterval, h(q).number
   PRINT , h(q).classinterval, h(q).number
   q = q + 1
LOOP UNTIL q = increment%

PRINT " "
PRINT USING "& & "; "Output went to file:  "; outname$

CLOSE

END
```